

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ВИСОКИХ ТЕХНОЛОГІЙ

В. о. завідувача кафедри квантової радіофізики

доц. Сергій КОЛЄНОВ

Протокол № \_\_\_\_ засідання кафедри

від “ \_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

**УДОСКОНАЛЕНА ЕЛЕКТРОННА СИСТЕМА  
КЕРУВАННЯМ БПЛА**

Випускна кваліфікаційна робота бакалавра

студента спеціальності

171 Електроніка

ОП “Електроніка (високі технології)”

**Соколовського Олександра Олександровича**

Науковий керівник

завідувач кафедри

теоретичних основ високих технологій

доцент, канд. техн. наук **Резніков Михайло Ігорович**

Оцінка захисту роботи

---

Київ – 2024 р.

## АНОТАЦІЯ

Соколовський О.О. Удосконалена електронна система керування БПЛА. – Випускна кваліфікаційна робота бакалавра за спеціальністю 171 Електроніка ОП “Електроніка (високі технології)”.

У роботі проведено розробку структурної та принципової схем, топології друкованої плати, програмного забезпечення польотного контролера, а також виготовлення експериментального зразка системи.

Встановлено, що використання 4 шарового стеку друкованої плати польотного контролеру в поєднанні з цифровими фільтрами обробки сигналів інерційної системи надає можливість коректної стабілізації та управління, що є критично важливим для безпечного та ефективного використання БПЛА в різних умовах. Отримані результати можуть бути використані для керування БПЛА в різних сферах їх застосування.

Ключові слова: БПЛА; польотний контролер; інерційна система; акселерометр; гіроскоп; кути Ейлера; ПІД корекція.

## ABSTRACT

Sokolovskyi O.O. Advanced electronic control system for UAVs. - Bachelor's thesis in the specialty 171 Electronics of the OP “Electronics (High Technologies)”.

The work involves the development of structural and circuit diagrams, printed circuit board topology, flight controller software, as well as the manufacture of an experimental sample of the system.

It has been established that the use of a 4-layer PCB stack of the flight controller in combination with digital filters for processing the signals of the inertial system makes it possible to correctly stabilize and control, which is critical for the safe and efficient use of UAVs in various conditions. The obtained results can be used to control UAVs in various applications.

Keywords: UAV; flight controller; inertial system; accelerometer; gyroscope; Euler angles; PID correction.

## ЗМІСТ

Скорочення і абривіатури .....	5
Вступ .....	6
РОЗДІЛ 1 ОБҐРУНТУВАННЯ СТВОРЕННЯ ПОЛЬОТНОГО КОНТРОЛЕРА ДЛЯ БПЛА.....	7
1.1 Загальні відомості .....	7
1.2 Обґрунтування необхідності.....	10
1.3 Огляд технічних рішень .....	11
РОЗДІЛ 2 РОЗРОБЛЕННЯ АПАРАТНОЇ ЧАСТИНИ.....	15
2.1 Розроблення структурної схеми польотного контролеру....	15
2.2 Вибір компонентів .....	16
2.3 Розроблення принципової схеми польотного контролеру ..	19
2.4 Особливості розміщення мікросхем на платі. ....	23
2.5 Розроблення РСВ дизайну друкованої плати польотного контролеру. ....	25
2.6 Монтаж і збирання дослідного зразка польотного контролеру. ....	30
РОЗДІЛ 3 РОЗРОБЛЕННЯ ПРОГРАМНОЇ ЧАСТИНИ.....	31
3.1 Аналіз та вибір IDE і мови програмування.....	31
3.2 Архітектура програмного забезпечення.....	34
3.2.1 MPU6050.....	37
3.2.2 QMC5883L.....	40
3.2.3 BMP280 .....	41
3.3 Розрахунки.....	44
3.4 Інтерфейс користувача .....	55

3.5	Протоколи роботи з зовнішніми модулями .....	57
3.5.1	CRSF .....	58
3.5.2	PWM .....	60
3.5.3	Dshot .....	62
3.5.4	SmartAudio/Tramp .....	65
3.5.5	OSD .....	67
РОЗДІЛ 4 ТЕСТУВАННЯ, РОЗРОБЛЕННЯ МЕТОДИКИ ЗАСТОСУВАННЯ .....		69
4.1	Тестування дослідного зразка польотного контролеру .....	69
4.2	Методика застосування .....	73
Висновки .....		77
Використані джерела .....		79

## СКОРОЧЕННЯ І АБРІВІАТУРИ

DShot -Digital Shot.

ESC - Electronic Speed Controller.

FPV - First Person View;

GPIO - general-purpose input/output;

IDE - Integrated Development Environment;

MEMS - Micro-electromechanical systems;

OSD - On Screen Display;

PCB - Printed Circuit Board;

PWM - pulse-width modulation;

UART - universal asynchronous receiver/transmitter;

VTX - video transmitter;

БПЛА - безпілотний літальний апарат;

## ВСТУП

У сучасному світі БПЛА знаходять все більше застосувань у різноманітних сферах діяльності людини. Від військових операцій та аграрного моніторингу до пошуково-рятувальних місій та розваг — БПЛА стають незамінними помічниками. Однак, ефективність та безпека використання БПЛА безпосередньо залежать від якості та надійності їх ключових компонентів, серед яких польотний контролер відіграє одну з головних ролей.

Мета кваліфікаційної роботи полягала у створенні удосконаленого прототипу системи керування БПЛА з підвищеною завадостійкістю плати польотного контролера та оптимізацією оброблення даних за допомогою цифрової фільтрації. Для досягнення поставленої мети у роботі успішно вирішено науково-технічне завдання, яке передбачало розроблення польотного контролера (синтез структурної та принципової схем, топології друкованої плати, відповідного програмного забезпечення), а також виготовлення експериментального зразка системи керування БПЛА в цілому.

У першому розділі роботи проводиться аналіз існуючих рішень у сфері польотних контролерів, їх переваг та недоліків, обґрунтування необхідності створення власного польотного контролера.

Другий розділ присвячений розробці апаратної частини польотного контролера, а саме розробці структурної схеми, вибору компонентів, розробці принципової схеми, розробленню РСВ дизайну.

Третій розділ присвячений розробці програмної частини польотного контролера, а саме розробці архітектури програмного забезпечення, отриманню та обробки вхідних даних та підготовки/передачі даних для зовнішніх модулів.

Четвертий розділ містить результати тестування та інструкцію використання.

## РОЗДІЛ 1 ОБҐРУНТУВАННЯ СТВОРЕННЯ ПОЛЬОТНОГО КОНТРОЛЕРА ДЛЯ БПЛА

### 1.1 Загальні відомості

БПЛА, які також відомі як дрони, є важливою частиною сучасних технологій в авіаційній сфері. Вони стали ключовими інструментами в різноманітних галузях. У аграрному секторі, вони використовуються для моніторингу посівів, розподілу добрив та захисту рослин від шкідників. У промисловості, БПЛА застосовують для інспекції та моніторингу великих об'єктів, що дозволяє підвищити безпеку та ефективність робочих процесів. У культурному секторі, дрони відкривають нові перспективи для кінематографії та фотографії, надаючи можливість створювати унікальний контент з небачених раніше ракурсів. У військовій сфері, БПЛА відіграють критичну роль, виконуючи різноманітні завдання, від розвідки та спостереження до точних ударів по ворожих об'єктах. Вони забезпечують передачу важливої інформації в реальному часі, коригування вогню артилерії, наведення авіації на цілі, а також можуть використовуватися для створення помилкових цілей та відволікання уваги противника.

БПЛА поділяються на кілька класів за типом конструкції літального апарату, кожен з яких має свої унікальні характеристики та області застосування.

Мультироторні системи, такі як квадрокоптери, октокоптери та гексакоптери, використовують кілька роторів для створення підйомної сили. Ці системи забезпечують високу маневреність та здатність зависати на місці, що робить їх ідеальними для завдань, які вимагають стаціонарного положення в повітрі, наприклад, при зйомці з висоти або інспекції об'єктів. Однак, через необхідність постійної роботи всіх роторів, мультироторні БПЛА мають обмежений час польоту та дальність, що обмежує їх використання у довготривалих місіях.

Криллаті системи, включаючи літаючі крила, моноплани та біплани, використовують аеродинамічну форму крила для створення підйомної сили. Це дозволяє їм здійснювати тривалі польоти на значні відстані, що робить їх підходящими для моніторингу великих територій, пошуково-рятувальних операцій та доставки вантажів. Недоліком крилатих БПЛА є їх нездатність зависати в повітрі та менша маневреність порівняно з мультироторними системами. Проте, завдяки сучасним технологіям, такі системи можуть бути оснащені спеціальними механізмами для покращення маневреності, наприклад, швидкодіючими елеронами та іншими аеродинамічними пристроями.

Гібридні системи поєднують в собі елементи обох класів, намагаючись збалансувати переваги мультироторних та крилатих БПЛА. Вони можуть мати здатність до вертикального зльоту та посадки, як у мультироторних систем, а також до тривалого польоту на великі відстані, як у крилатих систем. Це робить гібридні БПЛА дуже гнучкими та здатними виконувати широкий спектр завдань.

БПЛА поділяються на кілька класів за призначенням, кожен з яких має свої унікальні характеристики та області застосування.

Камікадзе-БПЛА — це невеликі, часто одноразові апарати, призначені для виконання самовідданих атак на ворожі цілі. Вони оснащені вибуховими речовинами та системами наведення для точного досягнення цілі. Ці БПЛА зазвичай використовуються для знищення важливих об'єктів або для створення психологічного ефекту на полі бою.

Розвідувальні БПЛА — це апарати, призначені для збору розвідувальної інформації про ворожі позиції, рух військ та інші стратегічно важливі дані. Вони можуть бути оснащені камерами високої роздільної здатності, сенсорами для виявлення хімічних речовин або радіації, а також засобами електронної розвідки для перехоплення комунікацій. Розвідувальні БПЛА зазвичай мають довгий час польоту та високу дальність дії.

Бомбардувальні БПЛА — це важкі апарати, призначені для доставки боєприпасів на великі відстані. Вони можуть нести керовані ракети, бомби або інші види озброєння. Ці БПЛА використовуються для точкових ударів по ворожих об'єктах, забезпечуючи підтримку наземних військ або виконуючи стратегічні завдання, такі як знищення інфраструктури противника.

БПЛА поділяються на кілька класів за радіусом застосування, кожен з яких має свої унікальні характеристики та області застосування.

Тактичні БПЛА — це апарати, призначені для виконання завдань на відносно коротких відстанях, зазвичай до кількох десятків кілометрів. Вони ідеально підходять для тактичної розвідки, спостереження та контролю за місцевістю, забезпечуючи оперативну інформацію наземним підрозділам.

Оперативно-тактичні або оперативні БПЛА мають більший радіус дії, що дозволяє їм виконувати завдання на відстанях до кількох сотень кілометрів. Ці БПЛА використовуються для оперативної розвідки та моніторингу, здатні забезпечувати підтримку в більш широкому оперативному просторі.

Стратегічні БПЛА призначені для дій на великі відстані, часто перевищуючи тисячі кілометрів. Вони можуть виконувати тривалі польоти і здійснювати стратегічну розвідку, електронне перехоплення, а також точкові удари по стратегічно важливих об'єктах противника. Ці БПЛА відіграють ключову роль у довготривалих місіях та операціях на великі відстані.

БПЛА поділяються на кілька класів за способом керування, кожен з яких має свої унікальні характеристики та області застосування.

FPV БПЛА — це апарати, які передають відео з камери, розташованої на борту, безпосередньо до очей оператора через спеціальні окуляри або монітор. Це дозволяє оператору керувати БПЛА з першої особи, ніби він знаходиться всередині апарату. FPV широко

використовується для гонок дронів, аерофотографії та інших застосувань, де необхідний високий рівень контролю та імерсії.

Автоматичні БПЛА — це апарати, які виконують польоти за допомогою автоматизованих систем, таких як інерціальні навігаційні системи або GPS. Вони можуть самостійно виконувати завдання за попередньо запрограмованим маршрутом, здійснювати зльоти та посадки, а також уникати перешкод. Автоматичні БПЛА ідеально підходять для моніторингу, картографування, сільськогосподарських застосувань та інших завдань, де потрібна висока точність та повторюваність.

Розвиток технологій БПЛА продовжує просуватися, і ми можемо очікувати появи нових класів та конструкцій, які будуть ще більш ефективними та адаптованими до специфічних потреб користувачів. Завдяки інноваціям у сфері матеріалознавства, електроніки та аеродинаміки, майбутнє БПЛА обіцяє бути захоплюючим та різноманітним.

## 1.2 Обґрунтування необхідності

Розробка власного контролера керування польотом БПЛА є стратегічно важливим завданням для України в сьогоденні з кількох причин:

- Технологічна незалежність: У сучасному світі, де технології швидко розвиваються, важливо мати можливість самостійно створювати та адаптувати технологічні рішення. Власний контролер дозволить Україні бути незалежною від іноземних постачальників та потенційних обмежень на експорт.
- Специфіка місцевих умов: Контролер, розроблений з урахуванням місцевих географічних та кліматичних умов, зможе краще виконувати завдання, специфічні для України. Це може включати

оптимізацію для польотів у різних погодних умовах та на різних висотах.

- Інновації та розвиток: Створення власного контролера сприятиме розвитку внутрішніх дослідницьких та виробничих потужностей. Це може стати поштовхом для інновацій у сфері авіаційних технологій та залучення молодих фахівців до наукової та технічної сфер.
- Безпека та захист даних: Використання власних контролерів з власним ПЗ забезпечує більший контроль над безпекою польотів та захистом даних. Це особливо важливо для військових та стратегічних застосувань, де конфіденційність та надійність є критичними.
- Економічна вигода: Розробка та виробництво контролерів в Україні може стимулювати економіку, створювати нові робочі місця та зменшувати залежність від імпорту.
- Відкритість для співпраці: Розробка контролера може відкрити двері для міжнародної співпраці та обміну технологіями, залучаючи інвестиції та партнерство з іншими країнами.

Створення власного контролера керування польотом БПЛА є не тільки важливим для забезпечення національної безпеки та незалежності, але й для підтримки сталого розвитку та інновацій в країні. Це дозволить Україні бути на передовій авіаційних технологій та забезпечить її місце у глобальному технологічному співтоваристві.

### 1.3 Огляд технічних рішень

Враховуючи важливість FPV дронів у сучасному світі, ринок польотних контролерів постійно розвивається, пропонуючи все більше варіантів для ентузіастів та професіоналів. Основні польотні контролери,

які зараз домінують на ринку, включають SpeedyBee F405, Matek Mateksys F411 та F405, Hakkerc F722, Betaflight F4, Foxeer F722, Jhemcu F4, та Diatone MAMBA F405. Після детального розгляду зазначених контролерів, можна виявити наступні схожості у їх апаратному виконанні:

- Всі вказані контролери використовують мікроконтролери компанії STM серії F4 або F7.
- У складі цих контролерів присутня майже ідентична сенсорна частина, зокрема, використання 6-осьового IMU BMI270 або ICM42688P.

Ці контролери мають декілька відомих проблем та недоліків:

- Проблеми зі стабільністю: Часто зустрічаються скарги на нестабільність плати, яка може “зависати” під час роботи в конфігураторі, особливо при підключенні батареї або перемиканні вкладок конфігуратора.
- Сумісність з приймачами: Існують випадки, коли користувачі зіткнулися з труднощами при налаштуванні роботи з приймачами деяких брендів, що може свідчити про проблеми сумісності або потребу в додаткових налаштуваннях.
- Нестабільність живлення: Спостерігається падіння напруги на платі, що може викликати зниження яскравості світлодіодів та може бути ознакою проблем з електропостачанням.
- Визнання акселерометра та гіроскопа: Іноді, після оновлення прошивки, акселерометр та гіроскоп не визнаються коректно, що вимагає додаткових налаштувань через CLI.
- Проблеми з прошивкою: Виникають складнощі при прошивці плати та налаштуванні через конфігуратор, що може вимагати додаткових зусиль та часу для вирішення.
- Завадостійкість: Основною проблемою цих моделей є їх компактні розміри, які не дозволяють дотримуватися рекомендованих

відстаней між мікросхемами MEMS та іншими компонентами, зазначених у технічних описах, що призводить до слабкої фільтрації електромагнітних завад і, як наслідок, до нестабільної роботи сенсорної системи.

Завдяки своїй сумісності з популярними програмними засобами керування, такими як ardupilot та betaflight, ці контролери пропонують велику гнучкість для користувачів. Вони можуть налаштовувати свої дрони для різноманітних завдань, від зйомки відео до участі в гонках. Однак, ця універсальність також призводить до певних викликів. Наявність великої кількості сумісних компонентів може ускладнити процес вибору оптимального обладнання, а також збільшити ризик збоїв через взаємодію різних систем.

Крім того, використання стандартного програмного забезпечення, такого як ardupilot та betaflight, може створювати потенційні ризики безпеки. Через однаковість програмних рішень після втрати контролю над дроном противник може легко перепрошити його та використовувати для своїх цілей, що може призвести до небажаних наслідків. Це робить важливим застосування унікального програмного забезпечення та захисних механізмів для забезпечення безпеки польотів.

З вище описаного можна зробити висновок, що не є край важливим якій саме контролер буде обраний за прототип розробленої системи. Всі вони складаються з однакових компонентів, керуються однаковим програмним забезпеченням, мають однакові проблеми з завадостійкістю. Тому розроблена схема не буде спиратися на вже існуючі схеми. За основу буде взятий більш потужний та дешевший мікроконтролер, 9 осьову IMU, розроблене власне програмне забезпечення з використанням цифрової (програмної) фільтрації. Вибір мікроконтролера, який не тільки економічно вигідний, але й має більшу потужність, може значно покращити продуктивність системи. Використання 9-осьового IMU

забезпечить більш точне відстеження руху, що є критично важливим для стабілізації квадрокоптера.

## РОЗДІЛ 2 РОЗРОБЛЕННЯ АПАРАТНОЇ ЧАСТИНИ

### 2.1 Розроблення структурної схеми польотного контролеру.

Основними завданнями польотного контролеру є:

- Отримання сигналів керування від пульта керування оператора;
- Отримання даних про положення БПЛА у просторі (обрахунок даних інерціальної системи);
- Обрахунок аеродинамічних параметрів для керування двигунами;
- Керування додатковими пристроями, що можуть бути встановлені на БПЛА.

На рисунку 2.1. зображено структурну схему польотного контролеру.



Рисунок 2.1 Структурна схема польотного контролеру

Дивлячись на рисунок польотний контролер базується на мікроконтролері ESP32, який обробляє сигнали від приймача (це може бути ELRS/CRSF або ін.), модуля сенсорів просторового положення БПЛА (інерціальна система). Оброблені сигнали мікроконтролером перетворюються в сигнали керування периферійними засобами через модуль сполучення. Це можуть бути сигнали PWM для керування сервоприводами або модулями ESC двигунів, Dshot1200 для керування цифровими модулями ESC двигунів, UART для керування модулями GPS, ініціації, OSD, SmartAudio/Tramp для керування відео передавачами VTX.

## 2.2 Вибір компонентів

Польотний контролер буде базуватися на мікроконтролері ESP32, гіроскопі MPU6050, магнітометрі QMC5883L, барометрі BMP280.

ESP32 — це серія мікроконтролерів типу “система на кристалі”, які мають інтегровані контролери Wi-Fi і Bluetooth (дворежимний), низьке енергоспоживання і невисоку ціну.

- Процесор: Xtensa двоядерний 32-розрядний LX6 мікропроцесор, що працює на 200 або 240 МГц і виконує до 600 DMIPS.
- Пам'ять: 520 Кб пам'яті SRAM.
- Бездротовий зв'язок:
  - Wi-Fi: 802.11 b/g/N
  - Bluetooth: v4.2 BR/EDR і BLE
- Периферійні інтерфейси:
  - 12-розрядний АЦП до 18 каналів 8-біт ЦАПи
  - 10 × сенсорних датчиків
  - 1 × SPI
  - × I2S
  - × I2C
  - × UART
  - SD/SDIO/CE-ATA/MMC/eMMC хост-контролер
  - Ethernet MAC інтерфейс з підтримкою IEEE 1588
  - CAN bus 2.0
- Безпека:
  - Підтримка всіх функцій безпеки стандарту IEEE 802.11, включаючи WPA, WPA/WPA2 і WAPI
  - Безпечне завантаження
  - Шифрування флеш-пам'яті
  - Криптографічне апаратне прискорення: AES, SHA-2, RSA, ECC

- Управління живленням:
  - Внутрішній низький регулятор відключення
  - Індивідуальний енергетичний домен для RTC
  - 5 мкА струм режиму “глибокий сон”

Порівняно з найпоширенішими мікроконтролерами в цій сфері а саме stm32 серії f4, esp32 має менше GPIO, більшу частоту процесору (у stm32 f4 серії 168 МГц), та більшу обчислювальну потужність (210DMIPS у stm), та нижчу ціну. Крім цього, великою перевагою є наявність Wi-Fi.

MPU-6050 - це інтегрований 6-осьовий сенсор руху, який поєднує 3-осьовий гіроскоп і 3-осьовий акселерометр на одному кремнієвому кристалі, разом з вбудованим Цифровим Процесором Руху (DMP), який обробляє складні алгоритми MotionFusion.

Деякі ключові характеристики MPU-6050:

- 3-осьовий гіроскоп і 3-осьовий акселерометр на одному чіпі.
- Різні діапазони чутливості: гіроскоп має програмований діапазон повної шкали  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$ , і  $\pm 2000$  °/сек, а акселерометр має діапазон  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$ , і  $\pm 16g$ .
- Вбудований температурний сенсор.
- Внутрішній осцилятор з варіацією  $\pm 1\%$  протягом робочого температурного діапазону.
- Інтерфейс I2C для зв'язку.

MPU-6050 також може підключатися до зовнішніх магнітометрів або інших сенсорів через додаткову майстерну шину I<sup>2</sup>C, дозволяючи пристрою збирати повний набір даних сенсорів без втручання системного процесора.

Порівняно з собі подібними MPU6050 відрізняється достатньою точністю, стабільністю, меншим енергоспоживанням.

QMC5883L - це багаточіповий тривимірний магнітний сенсор, який інтегрує магнітні сенсори з ASIC (application-specific integrated circuit) для обробки сигналів.

Деякі ключові характеристики QMC5883L:

- 3-осьові магніторезистивні сенсори в корпусі LGA розміром 3x3x0.9 мм, що гарантують роботу в широкому діапазоні температур від -40 °C до +85 °C.
- 16-бітний АЦП з низьким рівнем шуму, що досягає роздільної здатності поля в 2 мілі-Гаусс.
- Широкий діапазон магнітного поля ( $\pm 8$  Гаусс), що максимізує повний динамічний діапазон та роздільну здатність сенсора.
- Температурно компенсований вихід даних та вихід температури.
- Інтерфейс I2C зі стандартними та швидкими режимами.
- Максимальна частота виходу даних 200 Гц.
- Широкий діапазон робочих напруг (2.16В до 3.6В) та низьке споживання енергії (75 мкА).

BMP280 - це абсолютний барометричний тискосенсор, який особливо підходить для мобільних застосувань.

Деякі ключові характеристики BMP280:

- Діапазон роботи: тиск 300 - 1100 hPa, температура -40 - 85°C.
- Абсолютна точність: приблизно  $\pm 1$  hPa.
- Відносна точність:  $\pm 0.12$  hPa (типово), що еквівалентно  $\pm 1$  метру.
- Середнє споживання струму: 2.74  $\mu$ А.
- Час вимірювання: 5.5 мс.
- Напруга живлення: VDDIO 1.2 - 3.6 V, VDD 1.71 - 3.6 V.
- Роздільність даних: тиск 0.01 hPa ( $< 10$  см), температура 0.01° C.
- Інтерфейс I2C зі стандартними та швидкими режимами.

## 2.3 Розроблення принципової схеми польотного контролеру

Розробку схеми польотного контролеру, як і подальшу роботу над апаратною частиною цього проекту здійснював у AltiumDesigner. Оскільки апаратна частина польотного контролеру включає декілька різних модулів, а також модулі, що часто повторюються в даному проекті було прийнято рішення створювати ієрархічний проект.

Принципові схеми блока живлення, модуля перетворювачів рівнів, модуля сенсорів, а також блока мікроконтролера наведені нижче.

На рисунку 2.2 зображена принципова схема розробленого польотного контролеру.

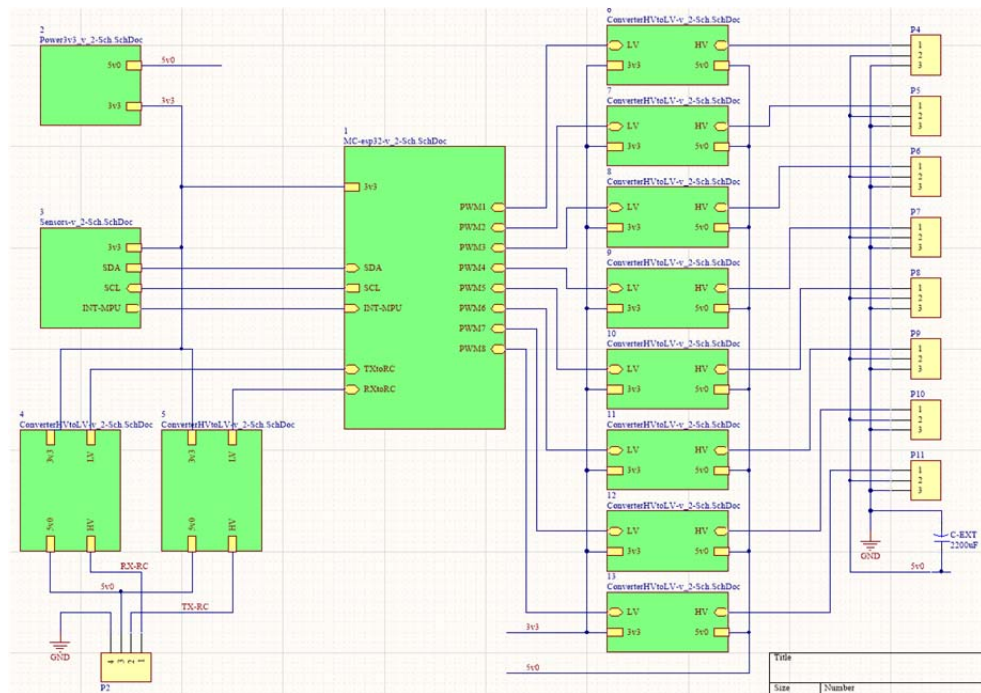


Рисунок 2.2 Принципова схема польотного контролеру

Як видно з рисунку, польотний контролер складається з мікроконтролеру ESP32 на який на ноги IO16 та IO17 (RXtoRC, TXtoRC) через конвертори рівнів подаються сигнали RX та TX від зовнішнього приймача. На ноги IO21, IO22, IO4 подається сигнал шини I2C по якій циркулюють дані від модуля сенсорів. Оброблені сигнали за відповідними

сценаріями через конвертори рівнів подаються на вихідні роз'єми для периферійних засобів.

На рисунку 2.3 зображено принципову схему блоку живлення.

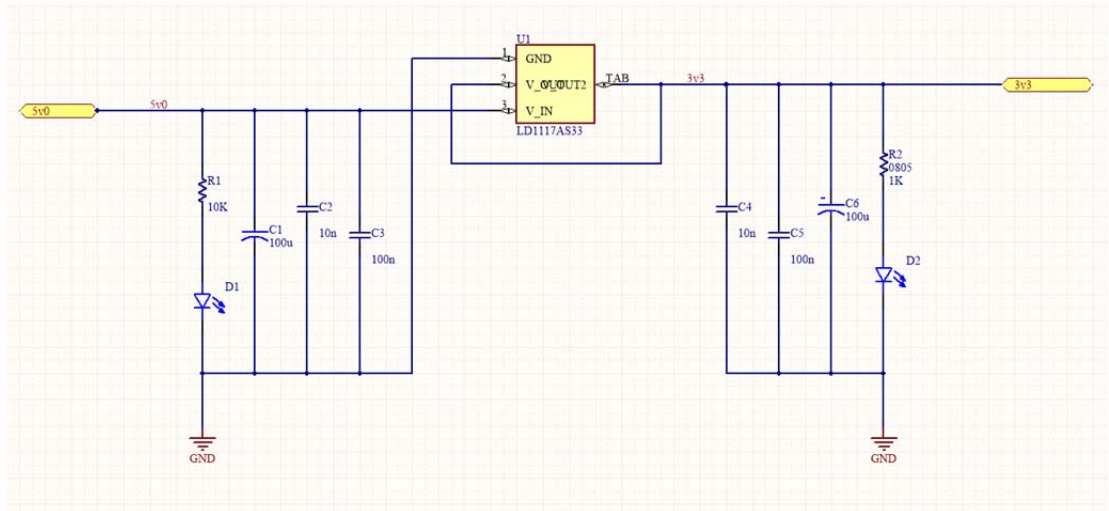


Рисунок 2.3. Принципова схема блоку живлення

Як видно з рисунку блок живлення складається з ланцюгів візуальної сигналізації наявності входної/вихідної напруги (R1 D1, R2 D2), фільтру входного/вихідного живлення (C1 C2 C3, C4 C5 C6). Враховуючи, що максимальна частота елементів польотного контролеру 240МГц, струм живлення коливається від 0,6А до 1А, падіння напруги при роботі мікроконтролера не перевищує 50мВ, то ємність конденсаторів фільтрів живлення буде дорівнювати 100нF(C3 C5) та 10нF(C2 C4). Електролітичні конденсатори C1, C6 слугують для фільтрації низької частоти та зберігання значних об'ємів енергії вибираємо плівкового типу з ємністю 100uF. Розрахунок проводився за формулою визначення ємності фільтрації пульсацій напруги.

$$C \approx \frac{I}{f \Delta V}$$

де:

$C$  – ємність конденсатора (Ф),

$I$  – струм живлення (А),

$f$  - частота пульсації (Гц),

$\Delta V$  – допустиме падіння напруги (В).

Формула 2.1 Формула ємності фільтрації пульсацій напруги

На рисунку 2.4 зображено модуль перетворювача рівнів напруги.

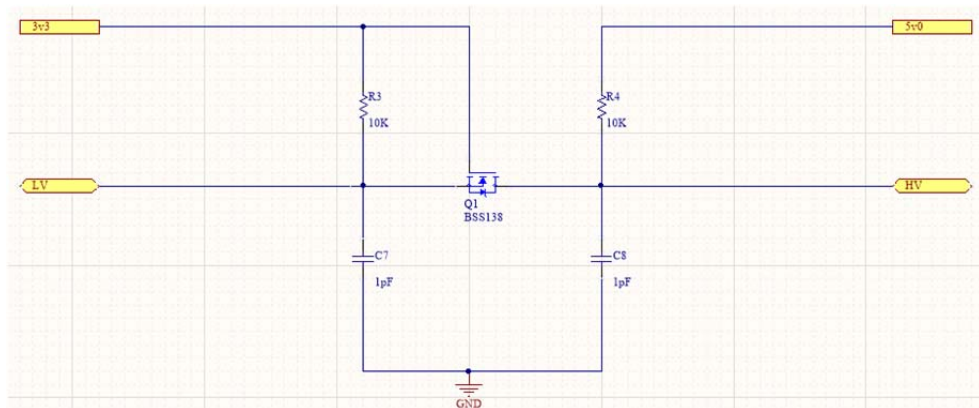


Рисунок 2.4. Модуль перетворювача рівнів напруги

Як видно з малюнку, модуль перетворювача напруги складається з польового N-канального транзистора BSS138, фільтруючих ємностей та захисних резисторів. У початковому стані жодна із сторін LV HV не підтягує лінію зв'язку до “0”, на стороні LV лінія підтягнута до “1” резистором R3, на стороні HV лінія підтягнута до “1” резистором R4. Gate і Source транзистора мають однаковий потенціал 3.3В, транзистор закритий. Таким чином, на вході обох сторін LV HV присутня логічна “1”, для кожного зі своїм рівнем напруги. В наступному стані сторона LV встановлює значення “0”. Source транзистора приймає низький потенціал, тоді як на Gate залишається 3.3В. Транзистор відкривається, сторона HV підтягується до низького рівня через відкритий транзистор, на стороні HV пристрою встановлюється логічний “0”. В останньому стані сторона HV встановлює на своїй стороні “0”. Через діодний перехід Source-Substrate

транзистора напруга на Source знижується доти, доки напруга на Gate не перейде пороговий рівень, потім транзистор відкривається, і сторона LV пристрою підтягується до “0” транзистором, що відкрився.

На рисунку 2.5 зображено модуль сенсорів.

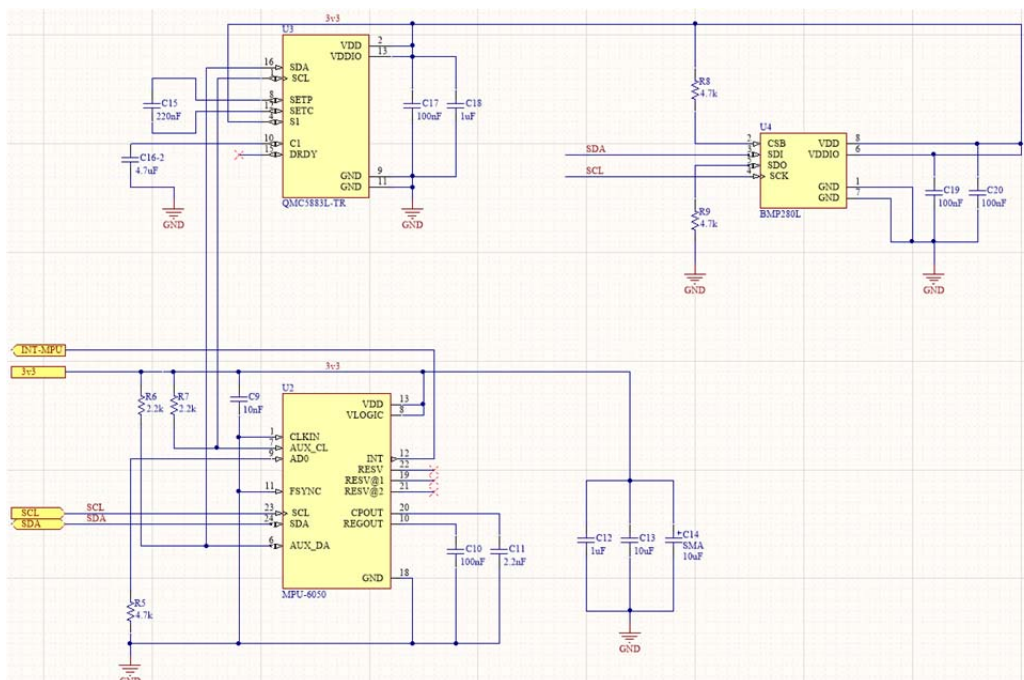


Рисунок 2.5. Модуль сенсорів

Як видно з рисунку, модуль сенсорів складається з 6-осьового акселерометра/гіроскопа MPU6050, магнітометра QMC5883L, барометра BMP280L та об'язувальних елементів, визначених даташитами. Принципова схема та топологія друкованої плати модуля сенсорів розроблені так, що замість MPU6050 можливо використовувати MPU6000 та MPU6500. Замість QMC5883L можливо використовувати HMC5883L. Замість BMP280L можливо використовувати BME280L. Всі ці мікросхеми зв'язані між собою та мікроконтролером за допомогою шини передачі даних I2C. Принцип роботи шини I2C, методи розміщення підтягуючих резисторів R15, R16, вибір їх номіналу визначені в специфікації “The I2C-bus and how to use it”[1].

На рисунку 2.6 зображено блок мікроконтролера.

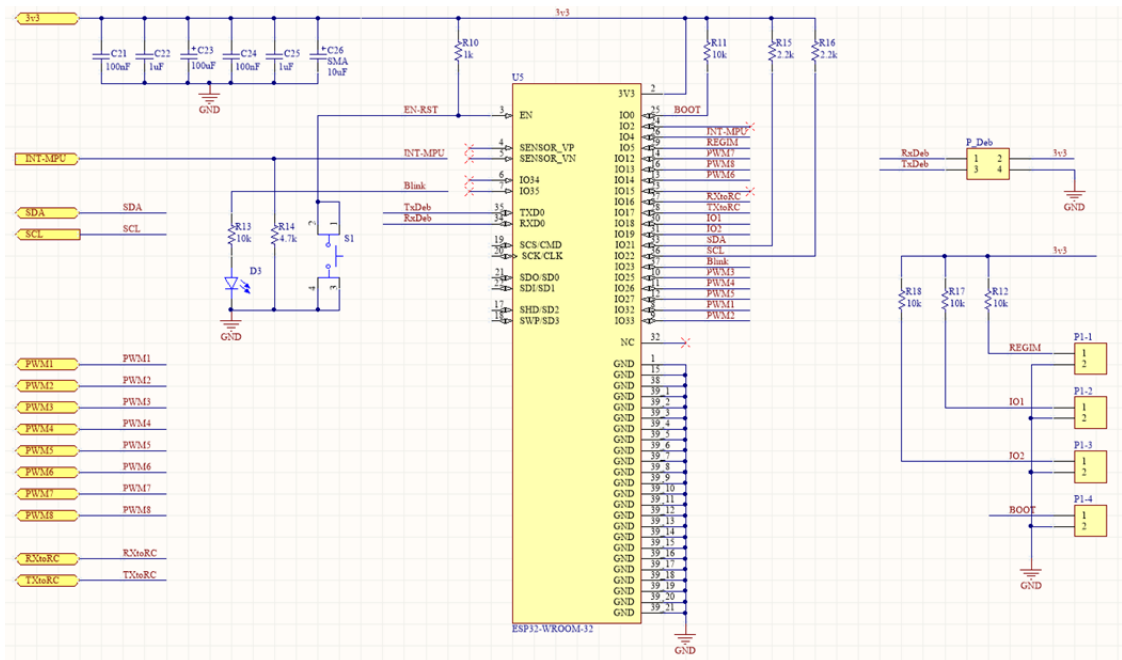


Рисунок 2.6. Блок мікроконтролера

Як видно з рисунку, блок мікроконтролера складається з:

- мікроконтролера ESP32 і обв'язувальних елементів (R10 – enable, R11 - boot). Данні ноги за допомогою зазначених резисторів підтягуються до +3,3В;
- ланцюга ідентифікуючого світлодіода (R13 D3);
- підтягуючих резисторів до +3,3В для лінії передачі переривання від модуля сенсорів, шини I2C, а також пінів керування режимами (R12, R14 - R18);
- блоку фільтрування живлення.

## 2.4 Особливості розміщення мікросхем на платі.

В ході розробки принципової схеми польотного контролеру були вивчені даташити ESP32, MPU6050, QMC5883L, BMP280, а також BSS138, LD1117AS33 та визначені особливості їх розміщення на друкованій платі.

В результаті вивчення даташитів можливо зробити наступні висновки (даташити компонентів наведені у додатку):

- ESP32 повинен розміщуватись так, щоб відстані від антени до металевих предметів корпусу були не менше як 15 мм. Крім цього, під самою антенною на друкованій платі не повинно бути жодних ланцюгів сигналу або шарів (полігонів) заземлення чи живлення. В розділі Power supply даташиту зазначено що модуль ESP32 вже має захисні конденсатори та індуктивності в своєму складі. Однак рекомендується додатково встановити ємності 0.1 мкФ на ланцюги живлення. Оскільки модуль ESP32 буде використовуватись в пілотному контролері дрону з електродвигунами, то можливі різноамплітудні та різночастотні завади по живленню та заземленню. Для запобігання цього будемо використовувати ємності з номіналами 10 нФ та 100 нФ, а також танталовий електролітичний конденсатор з ємністю більше за 100 мкФ. Крім цього необхідно забезпечити підтяжку пін EN і IO0 до живлення[2];

- MPU6050 повинен розміщуватись так, щоб відстань від корпусу MPU6050 до великих елементів таких як корпус, кнопки, роз'єми, електролітичні конденсатори, дроселі була не менш ніж 6 мм. Траси, з'єднані з контактними площадками, повинні бути максимально симетричними. Дотримання максимальної симетрії і балансу при з'єднанні з площадками допоможе компоненту само вирівнюватися і призведе до кращого контролю зменшення кількості паяльної пасти після оплавлення. Щоб уникнути гармонічних зв'язків, даташитом заборонено прокладання активних сигналів в неекранованих сигнальних площинах безпосередньо під або над корпусом гіроскопа, а також заборонено прокладати траси або отвори під корпусом гіроскопа таким чином, щоб вони проходили під відкритою контактною площадкою[3];

- Крім того, що всі компоненти, які можуть містити чорні матеріали (нікель тощо), мають бути віддалені від QMC5883 з обох боків друкованої плати, також рекомендується, щоб під/поблизу датчика не було

жодної провідної мідної лінії в жодному з шарів друкованої плати. Зовнішній конденсатор C16 повинен бути керамічного типу з низьким значенням ESR. Точне значення ESR не є критичним, але рекомендується значення менше 200 мОм. Номінальна ємність конденсатора C16 становить 4,7 мкФ[4].

- BMP280 повинен розміщуватись так, щоб зазор над металевою кришкою був не менше 0.1 мм. Для корпусу пристрою необхідно забезпечити відповідну вентиляцію. Рідини не повинні безпосередньо контактувати з приладом. Пайку не можна проводити з використанням парофазних процесів, оскільки датчик може бути пошкоджений[5];

- До BSS138 та LD1117AS33 відсутні будь які особливості розміщення на друкованій платі[6, 7].

## 2.5 Розроблення PCB дизайну друкованої плати польотного контролеру.

На практиці за допомогою PCB Editor створив документ багатошарової друкованої плати. Враховуючи, що на друкованій платі є ланцюги заземлення (земля), живлення 5В та 3,3В, аналогові та цифрові ланцюги даних то така плата буде багатошаровою. Кількість шарів буде дорівнювати 4. За допомогою Layer Stack Manager визначаємо порядок розташування шарів та їх характеристики (товщина діелектрику, товщину припрегу, товщина міді). Стек даної друкованої плати наведено нижче.

На рисунку 2.7 зображено стек друкованої плати.

#	Name	Material	Type	Thickness
	Top Overlay		Overlay	
	Top Solder	Solder Resist	Solder Mask	0,01mm
1	Top Layer		Signal	0,036mm
	Dielectric 1	FR-4	Core	0,127mm
2	3v3		Plane	0,036mm
	Dielectric 3	Tg175	Prepreg	0,127mm
3	GND		Plane	0,036mm
	Dielectric 2	FR-4	Core	0,127mm
4	Bottom Layer		Signal	0,036mm
	Bottom Solder	Solder Resist	Solder Mask	0,01mm
	Bottom Overlay		Overlay	

Рисунок 2.7. Стек друкованої плати

З рисунку видно, що друкована плата має 4 шари Top Layer, 3v3, GND, Bottom Layer. Загальна товщина друкованої плати складає 0,544мм.

За допомогою PCB Rules and Constraints Editor налаштовуємо мінімальні відстані від перехідних отворів до ланцюгів, правила перетину ланцюгів, товщини ланцюгів живлення та сигнальних ланцюгів, топологію трасування, правила повороту ланцюгів, розміри перехідних отворів для ланцюгів живлення та сигнальних ланцюгів, правила для розташування шарів маски, а також правила до створення трафарету для нанесення паяльної пасти.

Враховуючи вимоги до розміщення компонентів, умови експлуатації, налаштування вимог Altium Designer топологія шарів приведена нижче.

На рисунку 2.8 зображено зовнішній вигляд шарів друкованої плати в вікні редактора PCB Editor Altium Designer.

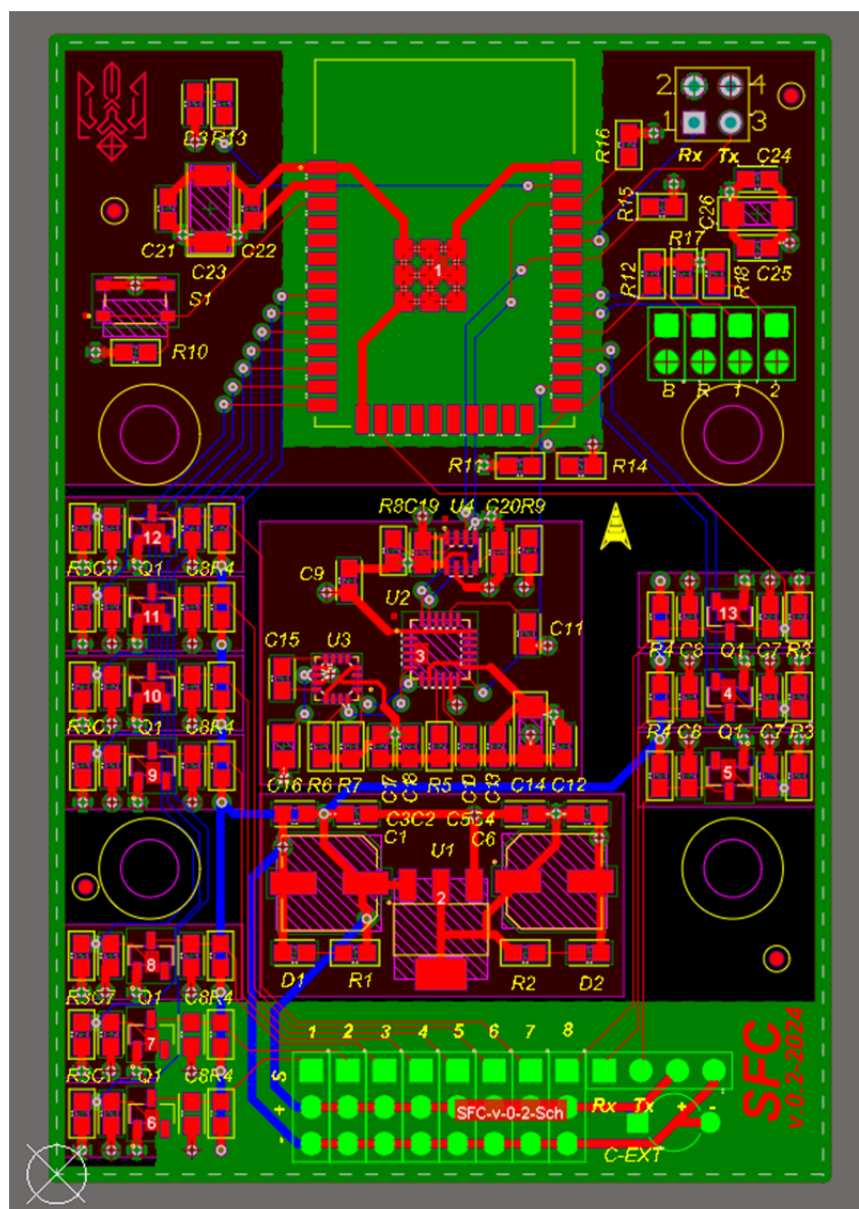


Рисунок 2.8 Зовнішній вигляд шарів друкованої плати в вікні редактора PCB Editor Altium Designer.

На даному рисунку зображено розташування компонентів на верхньому шарі TOP. На ньому розташовані компоненти десяти конверторів логічного рівня сигналу 3.3/5В, перетворювач постійної напруги з 5В у 3.3В, блок сенсорів (інерційна система), мікроконтролер ESP32 з необхідною обв'язкою. При розташуванні компонентів враховані мінімально допустимі відстані від компонентів MEMS та антени мікроконтролеру ESP32 до інших компонентів або з'єднувальних

ланцюгів. Для зменшення впливу завад на роботу мікроконтролера, а також мікросхем інерціальної системи ноги (піни) живлення та землі не підключені в безпосередній близькості до шарів живлення та землі. Вони підключені на деякій відстані через збільшені за розміром ланцюги живлення, а також групи конденсаторів.

Компоненти на нижньому шарі Bottom не розміщуються. З'єднання компонентів здійснюється як на шарі TOP, так на шарі Bottom.

Враховуючи, що є окремі живлення 3,3В та 5В необхідно зробити так, щоб ланцюги живлення 5В мінімально впливав на частину друкованої плати з мікроконтролером та сенсорами. Для цього ланцюги 5В розміщені на верхньому шарі лише в частині де вони необхідні (конвертори рівнів, джерело живлення). На рисунку 2.9 зображено верхній та нижній шари.

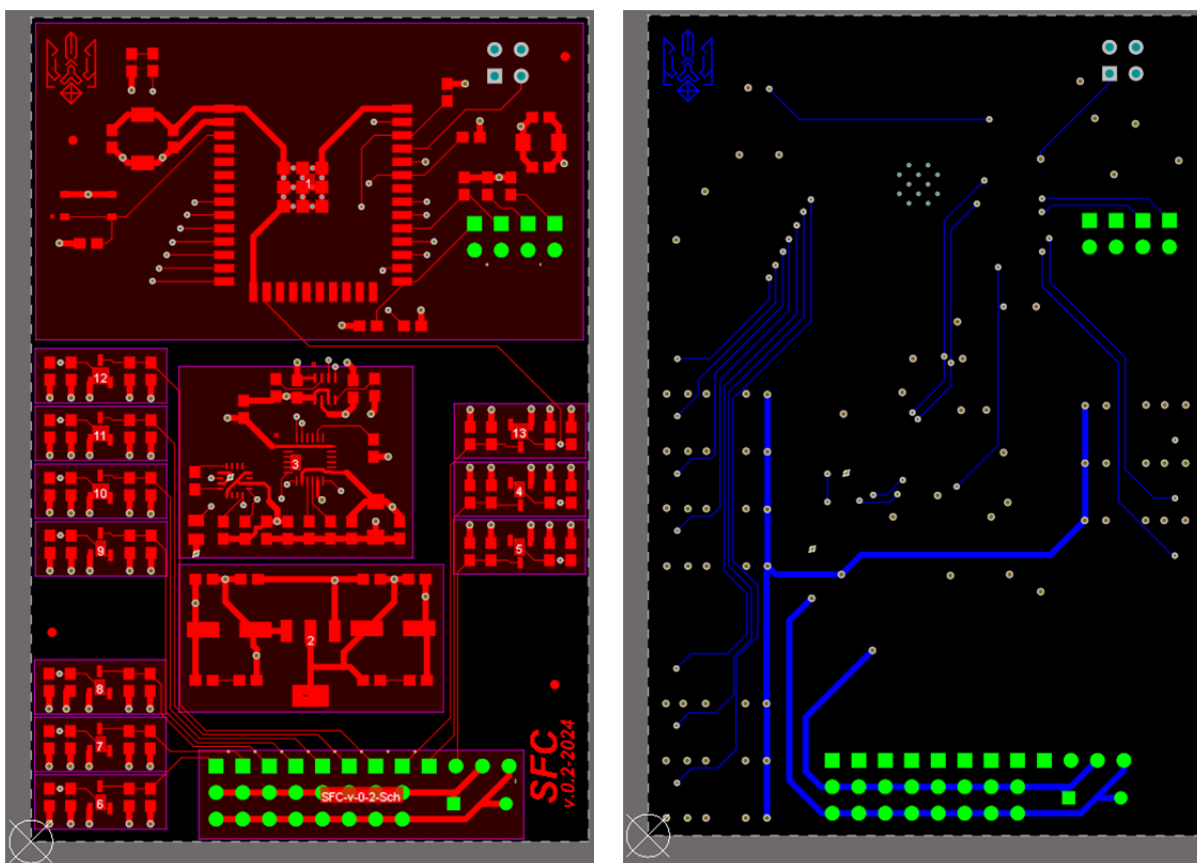


Рисунок 2.9 Верхній та нижній шари

З рисунку видно розташування контактних площадок елементів на друкованій платі, ланцюгів зв'язку.

На рисунку 2.10 зображено шар 3,3В та GND.

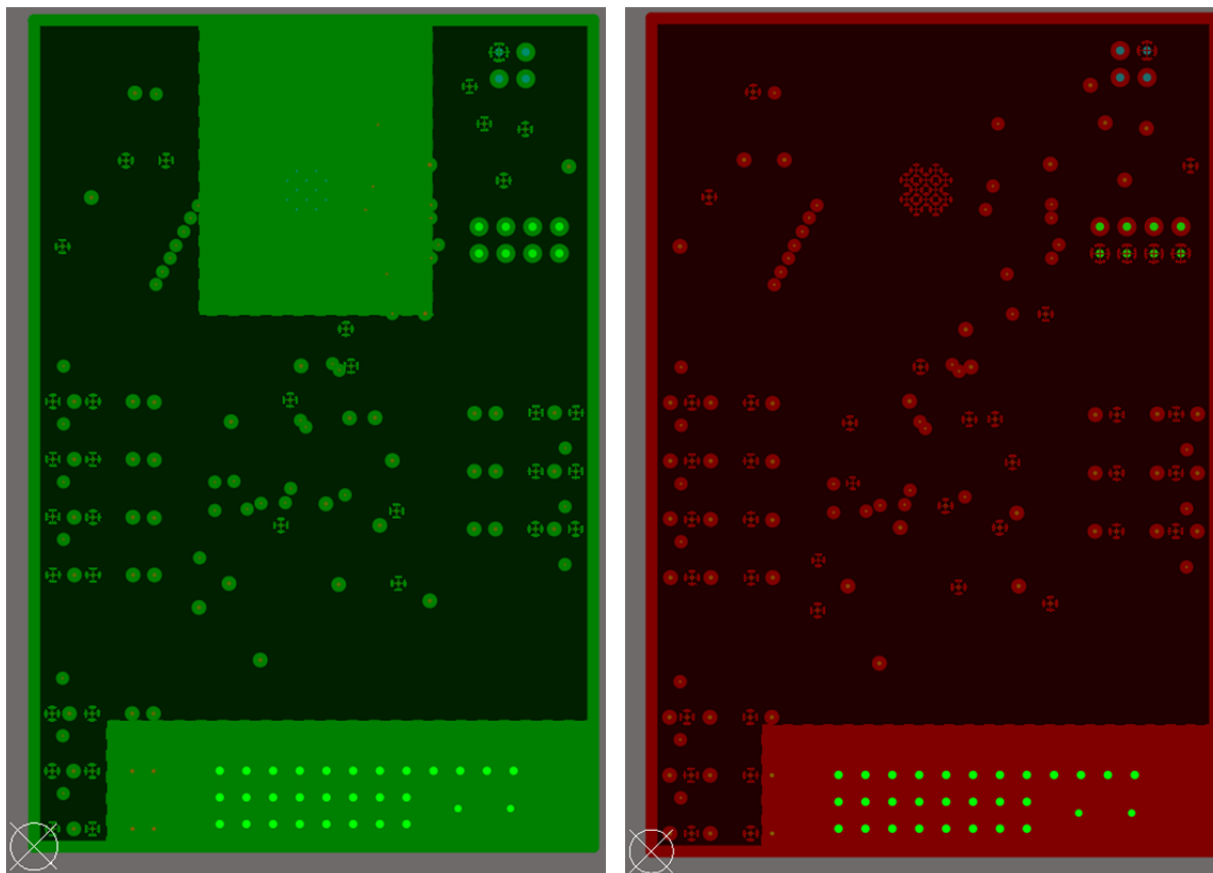


Рисунок 2.10. Шар 3.3В та GND

З рисунку видно геометричну форму шару 3,3В та GND. Також видно як на шарі 3,3В зроблено вивільнення простору під мікроконтролером ESP32.

За допомогою засобів Compile PCB Project та Design Rule Check перевіряємо правильність трасування, відповідність правилам та виправляємо помилки.

Враховуючи той факт, що для виробництва друкованих плат необхідні інші формати файлів ніж Schematic Editor та PCB Editor за допомогою інструменту імпорту файлів підготував файли Gerber. Крім цього сформував файл .BomDoc, де є повний перелік компонентів з їх

значеннями та посадковими місцями. Ці файли необхідні для передачі на виробництво друкованих плат і монтаж.

## 2.6 Монтаж і збирання дослідного зразка польотного контролеру.

Монтаж дослідного зразка проводив за допомогою паяльної станції з термофеном на монтажному столі.

Під час монтажу використовувався незмивний флюс та безсвинцева паяльна паста. Паста наносилась за допомогою шпателью через нержавіючий трафарет товщиною 0.127мм. Елементи на платі розставлялись вакуумним пінцетом.

## РОЗДІЛ 3 РОЗРОБЛЕННЯ ПРОГРАМНОЇ ЧАСТИНИ

### 3.1 Аналіз та вибір IDE і мови програмування

Вибір IDE та мови програмування є критично важливим для успішного виконання проекту, особливо в контексті вбудованих систем та мікроконтролерів. У даному аналізі розглянемо PlatformIO та мову програмування C++ у порівнянні з іншими популярними IDE та мовами програмування для вбудованих систем.

PlatformIO – це багатоплатформне інструментальне середовище для розробки програмного забезпечення для вбудованих систем. Вона підтримує безліч платформ, включаючи Arduino, ESP32, STM32, AVR та інші. Ключові особливості PlatformIO:

- Мультиплатформна підтримка: PlatformIO підтримує понад 800 плат з більш ніж 35 різних архітектур.
- Інтеграція з різними IDE: PlatformIO може використовуватися з різними середовищами розробки, такими як VS Code, Atom, CLion та інші.
- Автоматизація робочих процесів: Платформа надає можливості автоматизації збірки, тестування та розгортання програмного забезпечення.
- Пакетний менеджер: PlatformIO включає потужний менеджер бібліотек, що спрощує керування залежностями.

Таблиця 3.1

#### Порівняння IDE

Назва IDE	Переваги	Недоліки
Arduino IDE	Простота у використанні. Велика кількість готових бібліотек.	Обмежена функціональність порівняно з PlatformIO.

	Підходить для початківців.	Відсутність підтримки великої кількості платформ.
Eclipse з ESP-IDF	Потужне IDE з розширеними можливостями налагодження та управління проектами. Повна підтримка ESP-IDF, офіційного середовища розробки для ESP32.	Може бути складним у налаштуванні та використанні, вимагає багато часу для навчання і вимагає більше системних ресурсів.
Visual Studio з розширенням ESP-IDF	Широкі можливості налаштування, великі розширення, хороші налагоджувальні можливості, добре інтегрується з PlatformIO.	потребує налаштування, не така легка, як деякі альтернативи.
MPLAB X IDE	Потужний інструментарій для роботи з мікроконтролерами Microchip. Інтеграція з апаратними налагоджувачами.	Обмежена підтримка інших платформ. Вища складність для новачків. Не підтримує ESP32
Keil MDK	Оптимізоване для ARM мікроконтролерів. Висока продуктивність та багатифункціональність.	Висока вартість ліцензії. Обмеження у безкоштовній версії. Не підтримує ESP32.

C++ – це мова програмування з високим рівнем контролю над апаратним забезпеченням, яка широко використовується для розробки вбудованих систем. Основні переваги використання C++:

- Висока продуктивність: C++ дозволяє написати дуже ефективний і швидкий код.

- Контроль над апаратним забезпеченням: С++ дозволяє безпосередньо працювати з апаратними ресурсами, що важливо для вбудованих систем.
- Широка підтримка: Існує велика кількість бібліотек та інструментів для розробки на С++.

Таблиця 3.2

### Порівняння мов програмування

Назва мови програмування	Переваги	Недоліки
Python	Легко навчитися та використовувати. Велика кількість бібліотек для різних задач.	Менша продуктивність порівняно з С++. Вищі вимоги до ресурсів.
MicroPython	Легше писати та налагоджувати, широка підтримка швидкого створення прототипів, чудово підходить для початківців.	повільніша швидкість виконання, більше використання пам'яті, обмежена продуктивність у реальному часі.
С	Висока продуктивність. Повний контроль над апаратним забезпеченням.	Відсутність об'єктно-орієнтованих можливостей. Вища складність у керуванні великими проектами.
Java	Портативність завдяки JVM. Хороші об'єктно-орієнтовані можливості.	Вища вимога до ресурсів. Низька продуктивність порівняно з С++.

PlatformIO та С++ забезпечують потужну комбінацію для розробки модуля контролю польоту БПЛА на ESP32. Вони пропонують необхідну

продуктивність, гнучкість та розширені можливості, необхідні для такого складного проекту. Порівняно з іншими IDE та мовами програмування, PlatformIO та C++ забезпечують правильний баланс між простотою використання, функціональністю та контролем, що робить їх ідеальним вибором для цієї програми.

### 3.2 Архітектура програмного забезпечення

Визначимо архітектуру програмного забезпечення польотного контролеру.

Основними завданнями польотного контролеру є:

- Отримання сигналів керування від пульта керування оператора (ELRS/CRSF) ;
- Отримання даних про положення БПЛА у просторі (обрахунок даних інерціальної системи, таких як MPU6050, QMC5883L, BMP280);
- Обрахунок аеродинамічних параметрів польоту БПЛА (розрахунок параметрів за кутами Ейлера, розрахунок параметрів за методом барометричного нівелювання, ПД корекція, PWM/Dshot для ESC);
- Керування додатковими пристроями, що можуть бути встановлені на БПЛА (UART, SmartAudio/Tramp) .

Виходячи з вище викладеного мікроконтролер ESP32 повинен реалізовувати фізичне зчитування/отримання вхідних даних від сенсорів та приймача, виконувати математичні задачі обчислення параметрів просторового положення БПЛА та поправочні коефіцієнти для забезпечення стабільного положення БПЛА, генерувати вихідні сигнали керування додатковими пристроями, необхідними для польоту БПЛА.

На рисунку 3.1 зображено структурну схему архітектури розробленого програмного забезпечення.

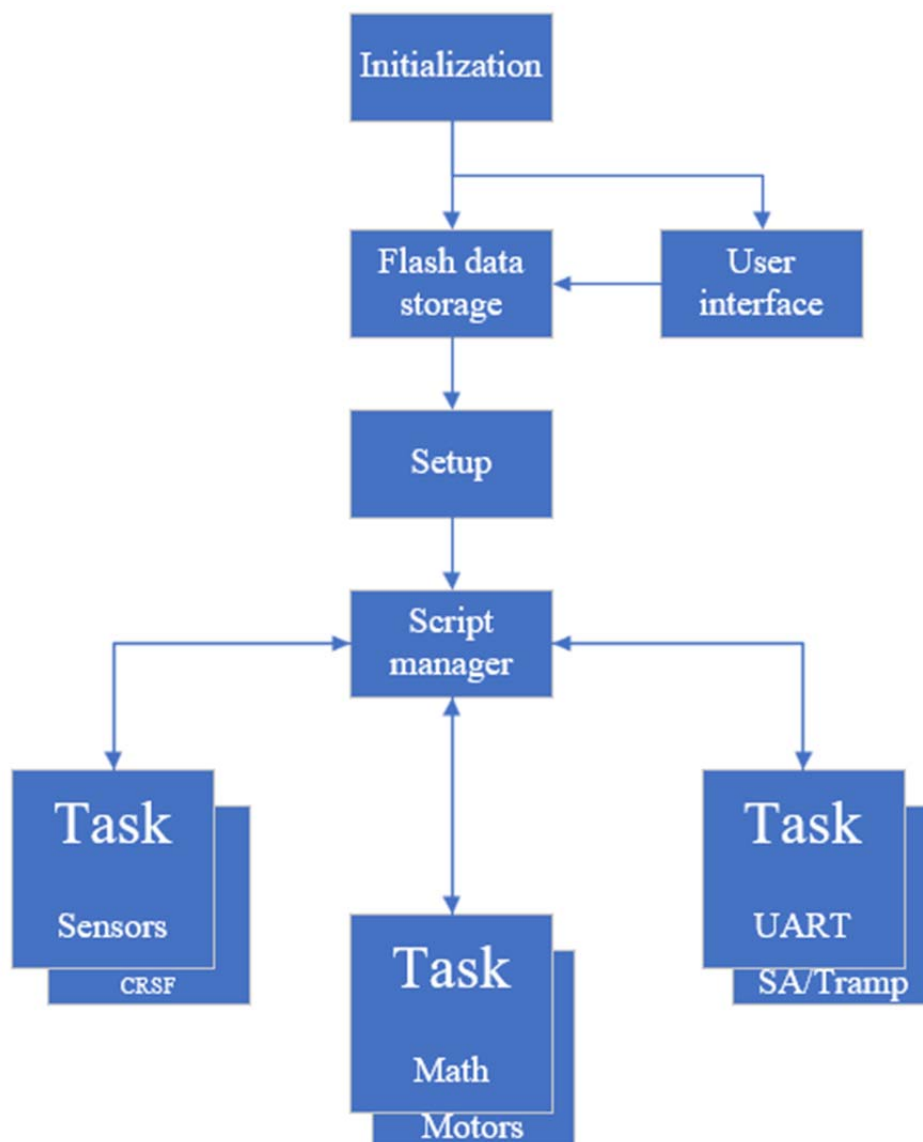


Рисунок 3.1 Структурна схема архітектури програмного забезпечення

Як видно з рисунку, першим етапом виконання програмного забезпечення є ініціалізація польотного контролера (ініціалізація бібліотек, об'єктів, змінних, портів вводу/виводу). Наступним етапом, в залежності від вибраного режиму, завантажується або інтерфейс користувача, або підпрограма читання параметрів керування БПЛА. Інтерфейс користувача

створює веб сторінку на якій можливо налаштувати параметри керування БПЛА та зберегти їх у флеш пам'яті контролера. Далі виконується встановлення параметрів та режимів функціонування програмного забезпечення. Останнім етапом є запуск завдань, які в залежності від сценарію БПЛА виконуються паралельно.

Паралельне виконання завдань здійснюється за допомогою бібліотеки FreeRTOS, та команди створення завдання

```
xTaskCreate(pvTaskCode, "pcName", usStackDepth, pvParameters, uxPriority, pvCreatedTask).
```

Однак, враховуючи переваги мікроконтролера ESP32, а саме наявність двох фізичних ядер, будемо використовувати наступну команду:

```
xTaskCreatePinnedToCore(pvTaskCode, "pcName", usStackDepth, uxPriority, pvCreatedTask, xCoreID)
```

де, pvTaskCode – вказівник на функцію, яка буде виконуватися в завданні;

pcName - ім'я завдання;

usStackDepth - розмір стека для завдання;

uxPriority - пріоритет завдання (число, починаючи з 0, де 0 - найменший пріоритет).

pvCreatedTask – вказівник на змінну, в яку буде записано дескриптор створеного завдання.

xCoreID - номер ядра, на якому завдання має виконуватися (0 або 1).

Виконання завдання оформлюється як:

```
void pvTaskCode (void* pvParameters) {
    for (;;) {
        xSemaphoreTake(mutexName, portMAX_DELAY);
        // Код для завдання
        // ...
        xSemaphoreGive(mutexName);
        vTaskDelay(time/portTICK_RATE_MS);
    }
}
```

```

    }
}

```

де, `pvParameters` - аргумент, який може бути використаний для передачі параметрів у задачу (якщо це необхідно).

`xSemaphoreTake(mutexName, portMAX_DELAY)`: Захоплює семафор (мьютекс) з ім'ям `mutexName`. Це зазвичай використовується для синхронізації доступу до загальних ресурсів (наприклад, критичної секції).

`xSemaphoreGive(mutexName)`: Звільняє семафор (мьютекс).

`vTaskDelay(time/portTICK_RATE_MS)`: Затримка виконання завдання на певний час (у мілісекундах).

### 3.2.1 MPU6050

MPU-6050 - це 6-осьовий пристрій відстеження руху, який поєднує в собі 3-осьовий гіроскоп, 3-осьовий акселерометр і цифровий процесор руху (DMP). Завдяки шині I2C він безпосередньо приймає вхідні дані від зовнішнього 3-осьового компасу QMC5883L, забезпечуючи повноцінний 9-осьовий вихід MotionFusion™. MPU-6050 також призначений для взаємодії з декількома неінерційними цифровими датчиками, такими як датчики тиску, через допоміжний порт I2C. MPU-6050 оснащений трьома 16-бітними аналого-цифровими перетворювачами (АЦП) для оцифрування виходів гіроскопа і трьома 16-бітними АЦП для оцифрування виходів акселерометра.

Для точного відстеження як швидких, так і повільних рухів компоненти мають програмований користувачем повномасштабний діапазон гіроскопа  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$  і  $\pm 2000^\circ/\text{сек}$  (dps) і програмований користувачем повномасштабний діапазон акселерометра  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$  і  $\pm 16g$ . Вбудований буфер FIFO на 1024 байт допомагає знизити енергоспоживання системи, дозволяючи системному процесору зчитувати

дані з датчиків серіями, а потім переходити в режим низького енергоспоживання, коли MPU збирає більше даних.

MPU-6050 має всі необхідні компоненти обробки та датчики на кристалі, необхідні для підтримки багатьох випадків використання, заснованих на русі, що робить його унікальним рішенням для енергозберігаючих додатків MotionInterface в портативних пристроях. Завдяки вбудованому виходу MotionFusion, DMP в MPU-6050 розвантажує системний процесор від інтенсивних обчислень, пов'язаних з обробкою руху, зводячи до мінімуму потребу в частому опитуванні виходу датчика руху. Зв'язок з усіма регістрами пристрою здійснюється за допомогою інтерфейсу I2C з частотою 400 кГц. Додаткові функції включають вбудований датчик температури і вбудований генератор з похибкою  $\pm 1\%$  в діапазоні робочих температур.

На рисунку 3.2 зображено структурну схему роботи бібліотеки MPU6050.

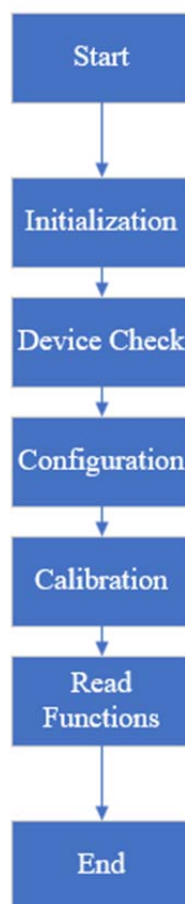


Рисунок 3.2 Структурна схема роботи бібліотеки MPU6050

Детальніше розберемо її роботу. Процес починається з виклику основної програми, яка використовує бібліотеку для роботи з датчиком MPU6050. На етапі ініціалізації (“begin”) відбувається встановлення адреси пристрою, ініціалізація шини I2C і налаштування різних параметрів сенсора, таких як джерело тактового сигналу та вимкнення режиму сну. Після ініціалізації виконується перевірка пристрою. Бібліотека читає регістр “WHO\_AM\_I” для перевірки, що під'єднано правильний пристрій і він відповідає на запити.

Далі йде налаштування параметрів чутливості та діапазону датчика функціями “setScale” та “setRange”. Функція “setScale” встановлює чутливість гіроскопа. Можливі значення включають 250DPS, 500DPS, 1000DPS і 2000DPS. Залежно від обраної шкали змінюється коефіцієнт переведення даних гіроскопа в градуси на секунду (DPS). Ці налаштування

записуються у відповідний регістр конфігурації гіроскопа. Функція “setRange” дає змогу вибрати діапазон вимірювань акселерометра: 2G, 4G, 8G або 16G. Залежно від обраного діапазону змінюється коефіцієнт переведення даних акселерометра в метри на секунду в квадраті ( $m/s^2$ ). Ці налаштування записуються у відповідний регістр конфігурації акселерометра.

Виконується функція калібровки гіроскопа “calibrateGyro” щоб компенсувати зміщення. Вона зчитує кілька зразків даних і обчислює середні значення для осей X, Y і Z, які потім використовуються для коригування подальших вимірювань.

Останній етап – читання даних. “readRawAccel” зчитує сирі дані по осях X, Y, Z; “readNormalizeAccel” переводить сирі дані у фізичні одиниці ( $m/s^2$ ). readRawGyro і readNormalizeGyro зчитують і нормалізують (градуси в секунду) дані гіроскопа відповідно .

### 3.2.2 QMC5883L

QMC5883L - це 3-осьовий магнітний датчик. Ця мікросхема невеликого розміру, що монтується на поверхню, має вбудовані магнітні датчики та призначена для високоточних застосувань, таких як компас, навігація та ігри в дронах, роботах, мобільних та персональних портативних пристроях. QMC5883L базується на найсучаснішій магніторезистивній технології з високою роздільною здатністю, ліцензований у Honeywell AMR. Разом зі спеціально розробленою 16-розрядною АЦП, вона пропонує переваги низького рівня шуму, високої точності, низького енергоспоживання, компенсації зсуву та температурної компенсації. QMC5883L забезпечує точність визначення компасного курсу від  $1^\circ$  до  $2^\circ$ . Послідовна шина I2C забезпечує простий інтерфейс. На рисунку 3.3 зображено структурну схему роботи бібліотеки QMC5883L.

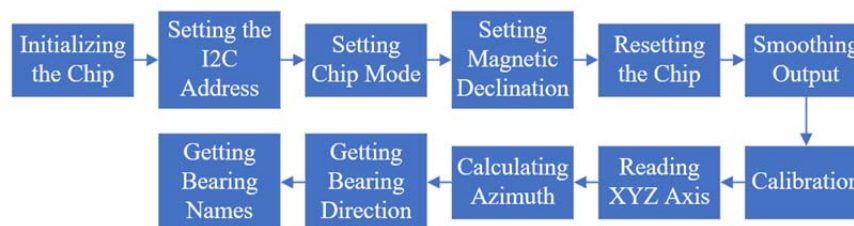


Рисунок 3.3 Структурна схема роботи бібліотеки QMC5883L

Детальніше розберемо її роботу. Першим етапом роботи бібліотеки є ініціалізації мікросхеми. Викликається функцію “init”, який виконує початкове налаштування мікросхеми QMC5883L, а саме ініціалізується шина I2C, встановлюються початкові регістри для ввімкнення мікросхеми і налаштування її режиму роботи. Встановлюється адреса I2C для мікросхеми. Ця адреса може бути налаштована користувачем, якщо вона відрізняється від стандартної. Далі налаштовується режим роботи мікросхеми, включно з режимом вимірювання, частотою оновлення даних, повним масштабом і коефіцієнтом передискретизації. Це виконується за допомогою функції `setMode()`. Встановлюється магнітне відмінювання для коригування показань азимута. Магнітне відмінювання залежить від географічного положення і може бути встановлене за допомогою функції `setMagneticDeclination()`. Згладжуються вихідні дані для зменшення шумів і отримання більш стабільних показань. У функції `_smoothing()` використовуються методи ковзаючого середнього і видалення крайніх значень. Функція `calibrate()` збирає дані з сенсора за певний період часу і встановлює калібровочні параметри. Функція `read()` виконує читання даних з мікросхеми та зберігає їх для наступної обробки.

### 3.2.3 BMP280

ВМР280 базується на технології п'єзорезистивного датчика тиску Bosch, що відрізняється високою електромагнітною стійкістю, високою точністю, лінійністю та довготривалою стабільністю. ВМР280 - це датчик абсолютного барометричного тиску, спеціально розроблений для мобільних застосувань, таких як: мобільні телефони, GPS-модулі або годинники. ВМР280 забезпечує високу продуктивність у всіх застосуваннях, що вимагають точного вимірювання тиску. ВМР280 працює з меншим рівнем шуму, підтримує нові режими фільтрації та інтерфейс I2C. ВМР280 має достатню точність  $\pm 0,12$  гПа, що еквівалентно  $\pm 1$  м перепаду висоти. Дуже низький температурний коефіцієнт зміщення (ТСО)  $1,5$  Па/К означає, що дрейф температури становить лише  $12,6$  см/К.

На рисунку 3.4 зображено структурну схему роботи бібліотеки ВМР280.

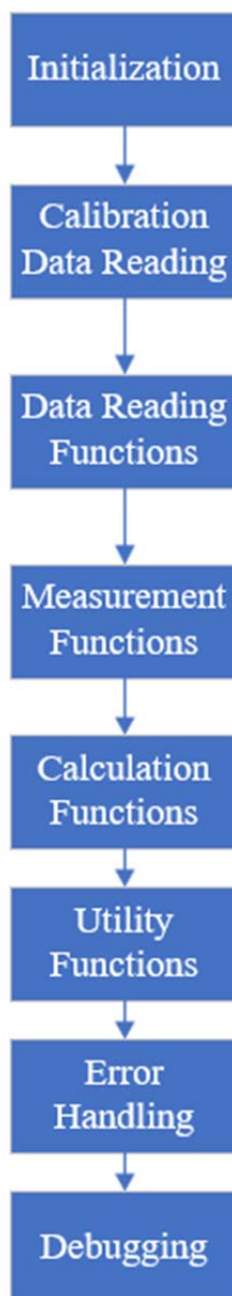


Рисунок 3.4 Структурна схема роботи бібліотеки BMP280

Детальніше розберемо її роботу. На першому етапі ініціалізації мікросхеми BMP280 викликається функцію “begin”, який виконує початкове налаштування, ініціалізацію шини I2C та зчитує калібрувальні дані з пристрою. Функція “readCalibration” зчитує калібрувальні дані з регістрів мікросхеми, що необхідні для коректних вимірювань температури та тиску.

Калібрувальні дані зчитуються з мікросхеми за допомогою функцій “readUInt” та “readInt”, які зчитують беззнакові та знакові цілочислені значення відповідно. Ці функції використовують I2C-шину для комунікації з мікросхемою.

Функція “setOversampling” дозволяє встановити коефіцієнт передискретизації, який впливає на точність та швидкість вимірювань. Функція “startMeasurement” починає цикл вимірювання, встановлюючи відповідні команди для мікросхеми залежно від коефіцієнта передискретизації.

Функція “getUnPT” отримує некалібровані значення тиску та температури з мікросхеми, які використовуються для подальших обчислень. Для обчислення скоригованих значень температури та тиску використовуються функції “calcTemperature” та “calcPressure”, які застосовують калібрувальні дані для коригування некаліброваних значень.

Функції “sealevel” та “altitude” використовуються для розрахунку тиску на рівні моря та висоти над рівнем моря відповідно, що дозволяє використовувати мікросхему BMP280 для метеорологічних вимірювань.

У разі виникнення помилок, функція “getError” повертає код помилки, який можна використовувати для діагностики проблем з I2C-комунікацією.

### 3.3 Розрахунки

Політ літального апарата (ЛА) в просторі є складним процесом, що вимагає врахування багатьох фізичних і технічних аспектів. Під час польоту польотний контролер керує органами керування для задання коректного вектору польоту. При цьому обраховуються наступні данні для польоту:

- Навігація: ЛА використовує системи GPS або інші навігаційні системи для визначення свого місцезнаходження та коригування маршруту.
- Стабілізація: Для підтримки стабільного польоту використовуються сенсори, такі як гіроскопи, акселерометри, магнітометри і барометри.
- Управління : Політ може бути автоматичним, керованим за допомогою попередньо запрограмованих маршрутів, або ручним, коли оператор контролює ЛА за допомогою радіозв'язку.

На рисунку 3.5 зображено основні осі орієнтації ЛА в просторі [10].

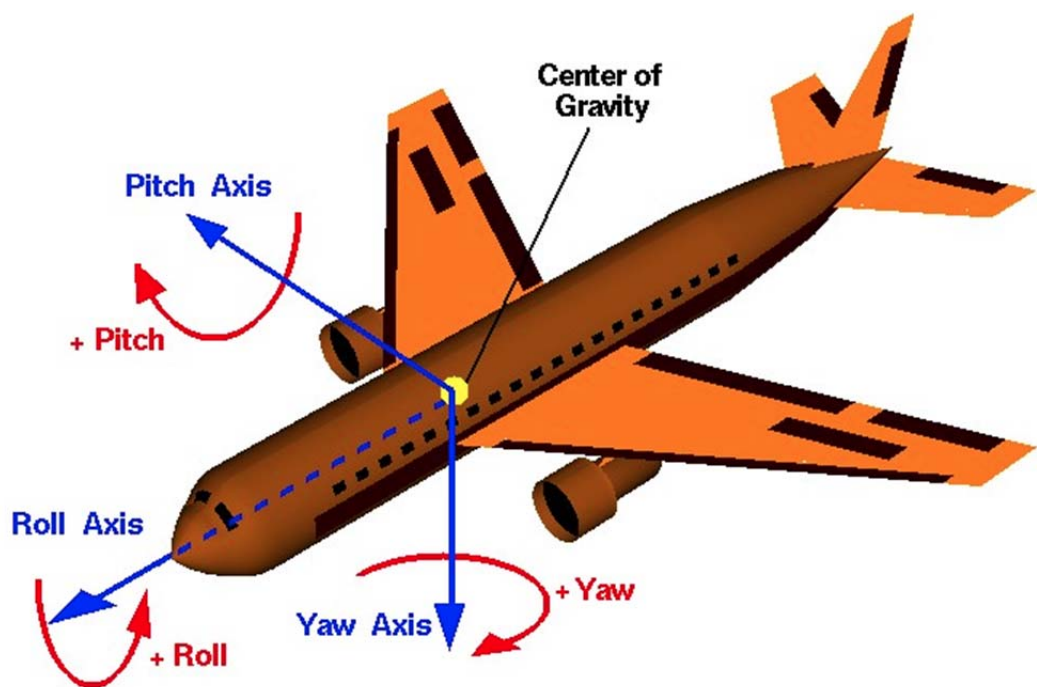


Рисунок 3.5 Основні осі орієнтації ЛА в просторі

З рисунку видно, що у ЛА є три основні осі орієнтації в просторі які складають між собою зв'язану систему координат. Літальний апарат під час польоту може вільно повертатися відносно трьох осей:

- Поздовжня вісь X(англ. roll), оберти довкола осі що проходить від хвоста до носа літака. Ці осі є системою координат і рухаються

разом з літаком, а також обертаються відносно поверхні землі разом із літаком.

- Поперечна вісь  $Y$  (англ. pitch), ніс літака рухається вниз або вгору довкола осі яка проходить від одного крила до іншого;
- Вертикальна вісь  $Z$  (англ. yaw), повороти вліво або вправо по курсу довкола осі яка проходить зверху вниз;

Ці повороти створюються за допомогою моментів сил, що виводять ЛА із рівноваги і змушують рухатись довкола основних осей. У БПЛА літакового типу, це відбувається завдяки повороту керуючих поверхонь, які змінюють розподіл загальної аеродинамічної сили відносно центру мас літака, у мультироторного типу це здійснюється за рахунок зміни швидкості обертання пропелерів двигунів. В авіаційній і авіаційно-космічній техніці власні обертання довкола цих трьох осей часто називаються кутами Ейлера.

На рисунку 3.6 зображено поєднання кутів Ейлера з власною системою координат та локальною системою координат схід-захід-верх.

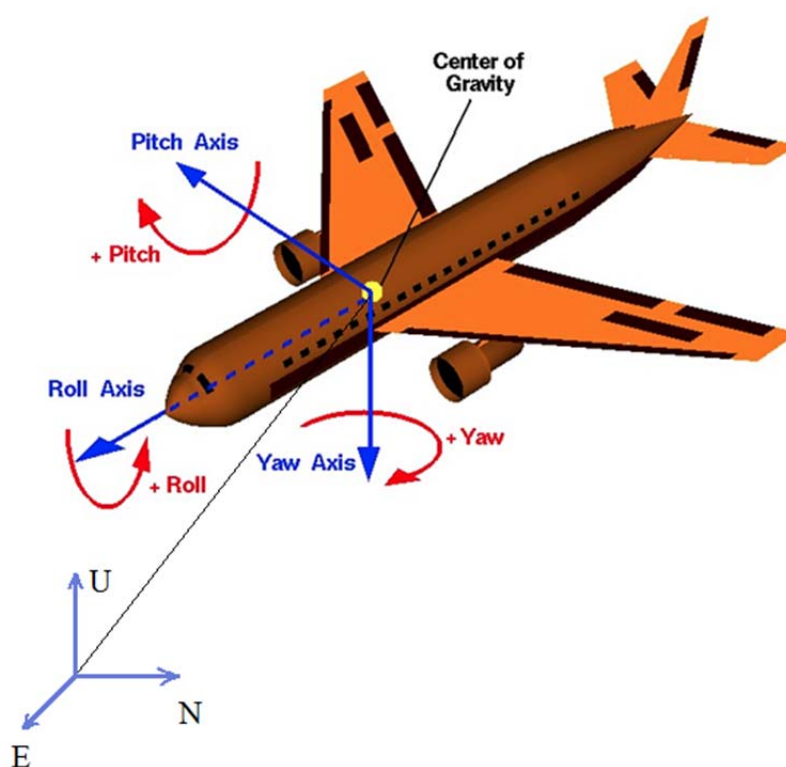


Рисунок 3.6 поєднання кутів Ейлера з власною системою координат та локальною системою координат схід-захід-верх.

З рисунка видно, що під кутами орієнтації ЛА кути Ейлера є roll, pitch, yaw, які пов'язують власну систему координат XYZ з локальною системою координат ENU. Кути roll, pitch, yaw позначають поворот, який необхідно здійснити осям XYZ щоб перейти в осі ENU. Відповідно, нульові кути означають, що вісь X об'єкту дивиться на схід, вісь Y об'єкту дивиться на північ, вісь Z до гори.

В польотному контролері за визначення осей pitch roll yaw та ENU відповідають акселерометр та магнітометр.

Акселерометр це датчик, що вимірює проекцію уявного прискорення на вісь чутливості. Уявного - тому що вимірює і силу тяжіння також, навіть тоді, коли акселерометр нерухомий. Найпростіше уявити акселерометр як грузик на пружинці, його вимірювання, що видаються, пропорційні ступеню розтягування пружини. Якщо акселерометр спочиває - пружина розтягнута лише силою тяжіння. Якщо прискорюється - то буде сума сил: інерції грузика ( $F = m\vec{a}$ ) і сили тяжіння ( $F_g = m\vec{g}$ ).

Прийmemo таку модель вимірювань тріади ортогональних (взаємно перпендикулярних) акселерометрів[8]:

$$a_{xyz} = m_a A_{xyz} + b_a + n_a$$

де  $a_{xyz}$  - вимірюване прискорення у власній системі координат XYZ,

$m_a$  - матриця перекоосу осей і масштабних коефіцієнтів акселерометра,

$A_{xyz}$  - вектор дійсного прискорення у власній системі координат XYZ,

$b_a$  - вектор зміщення нуля акселерометра,

$n_a$  - шум вимірювань.

Матриця перекосу осей і масштабних коефіцієнтів має такий вигляд[8]:

$$m_a = \begin{bmatrix} 1 + m_{a,1,1} & m_{a,1,2} & m_{a,1,3} \\ m_{a,2,1} & 1 + m_{a,2,2} & m_{a,2,3} \\ m_{a,3,1} & m_{a,3,2} & 1 + m_{a,3,3} \end{bmatrix}$$

де елементи, розташовані по головній діагоналі - це масштабні коефіцієнти та їхні похибки за трьома осями акселерометра, а інші елементи матриці - перекоси осей акселерометра.

Для акселерометра MPU6050 при ініціалізації вимірювальної системи будемо враховувати[3]:

- Зміщення нуля акселерометра - Zero-G Initial Calibration Tolerance ( $\pm 60mg$  для компонент X, Y,  $\pm 80mg$  для компоненти Z) для розрахунків переводим в одиниці g помножив на  $10^{-3}$ ;
- похибка масштабного коефіцієнта - Initial Tolerance ( $\pm 3\%$ ), що виражається у відсотках. Для розрахунків треба перевести в рази, помноживши на  $10^{-2}$ ;
- перекоси осей - Cross Axis Sensitivity ( $\pm 2\%$ ) також множимо на  $10^{-2}$ ;
- спектральна щільність потужності шуму акселерометра - Noise Power Spectral Density ( $300 \frac{\mu g}{\sqrt{Hz}}$ ) - переводимо чисельник у g помноживши все на  $10^{-6}$ ;

Магнітометр датчик, який вимірює проекцію індукції магнітного поля на вісь чутливості. Магнітометру властиві викривлення hard-iron і soft-iron. Hard-iron викривлення - це адитивний ефект, коли до вимірюваного поля додається постійна складова. Причиною може бути, наприклад, дія постійного магніту або власне зміщення нуля датчика. Викривлення soft-iron - мультиплікативний ефект, що відображає зміну напрямку та/або ослаблення вектора магнітної індукції. Цей ефект може бути спричинений наявністю металевго предмета в безпосередній

близькості від магнітометра або ж власними викривленнями датчика - похибкою масштабного коефіцієнта чи перекосом його осі чутливості.

Візьмемо модель вимірювань тріади магнітометрів[9]:

$$m_{xyz} = S_m M_{xyz} + b_m + n_m$$

де  $m_{xyz}$  вимірювання магнітометра у власній системі координат XYZ,

$S_m$  - - діагональна матриця перекоосу осей і масштабних коефіцієнтів (яка описує ефект soft-iron),

$M_{xyz}$  - вектор істинної магнітної індукції у власній системі координат,

$b_m$  - зміщення нулів магнітометра (описує ефект hard-iron),

$n_m$  - шум вимірювань.

Матриця перекоосу осей і масштабних коефіцієнтів магнітометра[9]:

$$S_m = \begin{bmatrix} 1 + S_{m,1,1} & S_{m,1,2} & S_{m,1,3} \\ S_{m,2,1} & 1 + S_{m,2,2} & S_{m,2,3} \\ S_{m,3,1} & S_{m,3,2} & 1 + S_{m,3,3} \end{bmatrix}$$

де елементи, розташовані на головній діагоналі - масштабні коефіцієнти та їхні похибки за трьома осями магнітометра, інші елементи матриці - перекооси осей магнітометра. Усі елементи матриці також враховують ефект soft-iron.

Для магнітометра BMP280 при ініціалізації вимірювальної системи будемо враховувати[4]:

- зміщення нуля -  $1\mu T$ ;
- похибка масштабних коефіцієнтів – 5%;
- перекооси осей -  $\pm 2\%$ ;
- шум на виході –  $0.6\mu T$ .

Завдяки наявності на Землі сили тяжіння, акселерометри “відчувають” напрямок вниз. Їхні вимірювання використовуються для розрахунку кутів крену і тангажу. Третій - кут нищпорення (а в даному

випадку - магнітного азимута), може бути визначений завдяки наявності у Землі магнітного поля. Вектор індукції магнітного поля вимірюється магнітометрами, і їхні вимірювання беруть участь у розрахунку кута нишпорення. Потрібно зазначити, що в розрахунку магнітного азимута використовуються вимірювання магнітометра, перераховані в площину[9].

$$\begin{aligned} roll &= atan\left(\frac{a_y}{a_z}\right) \\ pitch &= atan\left(\frac{-a_x}{\sqrt{a_y^2 + a_z^2}}\right) \\ yaw &= atan2\left(\frac{m_E}{m_N}\right) \end{aligned}$$

де  $atan2$  - функція повного арктангенса,

$a_x, a_y, a_z$  - вимірювання акселерометра за трьома осями у власній системі координат,

$m_E, m_N$  - вимірювання магнітометра по осях  $X', Y'$  (вимірювання магнітометрів перераховані в площину).

Для розрахунку висоти використовується метод барометричного нівелювання. Барометричне нівелювання – метод для визначення висотної різниці між двома точками на основі змін атмосферного тиску, зафіксованих у кожній точці. Оскільки атмосферний тиск залежить не лише від висоти, але й від погодних умов, таких як вологість повітря, цей метод найкраще працює, коли вимірювання проводяться швидко одне за одним і коли обидві точки знаходяться порівняно близько одна до одної. Існує доволі складна формула Лапласа:

$$h = 18401,2(1 + 0,00366t) \left(1 + 0,378 \frac{e}{p_0}\right) (1 + 0,0026 \cos 2\phi)(i + \beta h) \lg \frac{p_0}{ph}$$

Вона, крім температури і тиску, також враховує абсолютну вологість повітря  $e$  і широту  $\phi$  місця вимірювання, на практиці її використовують рідко, а використовують простішу формулу Бабіне:

$$h = 8000 \frac{2(\rho_0 - \rho h)}{\rho_0 + \rho h} (1 + at)$$

де  $\alpha$  – коефіцієнт розширення газів, дорівнює  $1/273$ ;

8000 – шкала геопотенційної висоти;

$p_0$  – тиск у точці, відносно якої проходить розрахунок;

$p_h$  – тиску у точці, висоту якої шукаємо

$t$  – температура.

Ці данні використовуються для корегування положення в просторі БПЛА. Однак, враховуючи наявність похибки обчислення для підтримки параметрів системи будемо використовувати пропорційно-інтегрально-диференціальний регулятор.

Він є одним із найпоширеніших автоматичних регуляторів та настільки універсальний, що застосовується практично в будь-якій галузі, де потрібне автоматичне керування. Наприклад, для регулювання температури в спеціальних печах, холодильниках, інкубаторах, паяльниках, соплах і столах 3D-принтерів, ІЧ-паяльних станціях та іншому обладнанні. Також його використовують для підтримки частоти обертів мотора у верстатах, в різноманітних балансуєчих пристроях, таких як гіроскутери, сігвеї, магнітні левітуючі платформи, а також квадрокоптерах і літаках з автопілотом.

ПІД-регулятор складається з трьох компонентів: пропорційної (P), інтегруючої (I) та диференціюючої (D) складових. Його вихідний сигнал формується як сума трьох значень, кожне з яких помножене на свій коефіцієнт. Ця сума після обчислень стає керуючим сигналом, який подається на виконавчий пристрій, позначимо його як *out*.

$$out = PkP + Iki + Dkd,$$

де  $k_P$ ,  $k_I$  і  $k_D$  – коефіцієнти, які треба налагодити для роботи ПІДа.

Значення можуть варіюватися від 0.001 до десятків і тисяч, залежно від конкретної системи. Коефіцієнт також може дорівнювати нулю, у цьому випадку відповідна компонента обнуляється. Таким чином, регулятор може бути налаштований як П, І, ПД та інші комбінації. Різні

системи вимагають різного підходу, саме тому ПД-регулятор такий універсальний.

Пропорційна складова являє собою різницю між поточним значенням з датчика та очікуваним значенням:

$$P = \textit{setpoint} - \textit{input}$$

де *setpoint* - очікуване значення,

*input* – значення з датчика.

Ця різниця називається помилкою регулювання, тобто показує, наскільки далеко система перебуває від заданого значення. Виходить, що чим більшою є помилка, тим більшим буде керуючий сигнал, і тим швидше система приводитиме керовану величину до заданого значення. Коефіцієнт  $k_P$  виконує роль підсилювача помилки і налаштовується вручну. Проте, коли система досягне заданого значення, помилка стане нульовою, і керуючий сигнал також зникне. Інакше кажучи, П-регулятор ніколи не зможе повністю усунути помилку, завжди залишатиметься певне відхилення. Пропорційна складова є основною в ПД-регуляторі і виконує найбільшу частину роботи; регулятор може працювати досить ефективно навіть лише з цією складовою. Пропорційна складова виправляє помилку в поточний момент часу.

Інтегральна складова накопичує помилку, підсумовуючи різницю між поточним і заданим значенням, помножену на період дискретизації системи (час, що минув від попереднього розрахунку  $dt$ ). Фактично, вона обчислює інтеграл від помилки за часом.

$$I = I + (\textit{setpoint} - \textit{input})dt$$

У самому регуляторі це ще множиться на коефіцієнт  $k_I$ , яким налаштовується інтенсивність цієї складової. В інтегральній складовій буквально накопичується помилка, що дозволяє регулятору з часом повністю її усунути, тобто привести систему точно до заданого значення з максимальною точністю. Інтегральна складова виправляє минулі, накопичені помилки.

Диференціальна складова визначається як різниця між поточною і попередньою помилкою, поділена на час між вимірами, тобто на  $dt$ , який є загальним періодом регулятора. Іншими словами, це похідна від помилки за часом.

$$err = setpoint - input$$

де  $err$  - помилка регулювання;

$$D = \frac{err - prevErr}{dt}$$

$$prevErr = err$$

Фактично, D-складова реагує на зміну сигналу з датчика, і чим сильнішою є ця зміна, тим більше значення додається до загальної суми. Іншими словами, D-складова дозволяє компенсувати різкі зміни в системі та, за правильного налаштування, запобігти сильному перерегулюванню і зменшити коливання. Коефіцієнт  $kD$  дозволяє налаштувати вагу або інтенсивність цієї компенсації, як і інші коефіцієнти регулюють свої складові. D-складова необхідна насамперед для швидких систем, тобто систем з різкими змінами, таких як квадрокоптери. D-складова виправляє можливі майбутні помилки, аналізуючи швидкість змін.

Для налаштування регулятора потрібно змінювати коефіцієнти:

- При збільшенні  $kP$  зростає швидкість досягнення встановленого значення і збільшується керуючий сигнал. Суто математично система не може прийти точно до заданого значення, оскільки під час наближення до нього пропорційна складова пропорційно зменшується. При подальшому збільшенні  $kP$  реальна система втрачає стійкість, що призводить до коливань.
- При збільшенні  $kI$  зростає швидкість компенсації накопиченої помилки, що дозволяє вивести систему точно до заданого значення з часом. Якщо система повільна, а  $kI$  надто великий, інтегральна сума може сильно зрости, що призведе до перерегулювання, яке може проявлятися як незатухаючі коливання з великим періодом.

Тому інтегральну суму в алгоритмі регулятора часто обмежують, щоб вона не могла збільшуватися і зменшуватися безмежно.

- При збільшенні  $kD$  зростає стабільність системи, оскільки вона не дозволяє системі змінюватися занадто швидко. Водночас, великий  $kD$  може стати причиною неадекватної поведінки системи та постійних стрибків керуючого сигналу, особливо якщо сигнал з датчика зашумлений. На кожну різку зміну сигналу з датчика  $D$ -складова реагуватиме зміною керуючого сигналу, тому сигнал з датчика потрібно фільтрувати.

Так як плата польотного контролера прикріплений до рами, інерційна система відчуває завади, які походять від: моторів, пропелерів, вітру на швидкість, загальний шум від рами, електроніки. Шуми від електроніки нівелюються конденсаторними фільтрами низьких та високих частот. Решту шумів потрібно фільтрувати цифровими методами.

Процес фільтрації полягає у видаленні зайвого шуму із сигналу від гіроскопа. На практиці з'ясовано, що частота шуму, а точніше вібрації достатньо низька та вимірюється в Гц. Вібрації створені рухом мультироторного БПЛА в залежності від швидкості знаходиться в межах 0-30 Гц. При значеннях вібрації від 30 Гц до 80 Гц квадрокоптер відчуває турбулентність від власних потоків. На практиці було визначено, що сигнал від гіроскопа містить інформацію про вібрацію від 0 Гц до 1000 Гц, але нас цікавить лише діапазон 0-80 Гц, оскільки це фактичний рух квадрокоптера, про який має знати ПІД-регулятор. Таким чином, було реалізоване рішення для фільтрації, щоб дозволити низьким частотам проходити через ПІД-регулятор, водночас послаблюючи високі частоти, і для цього ми можемо використовувати фільтр низьких частот.

В результаті використання методів викладених в цьому розділі отримали наступні результати:

- Розрахунок кутів по показникам акселерометра, гіроскопа, магнітометра здійснюється за 660-680 мкс. В порівнянні з Накгс F722 швидкість розрахунків зменшена втричі. У Накгс F722 вони складають 1980-2010 лише з обрахуванням показників від гіроскопа та акселерометра;
- Розрахунок ПІД регуляторів здійснюється за 960-980мкс. В порівнянні з Накгс F722 швидкість розрахунків зменшена в двічі (1920-1960мкс).

### 3.4 Інтерфейс користувача

Бібліотека інтерфейсу користувача дозволяє налаштовувати і керувати з'єднанням Wi-Fi та веб-сервером на мікроконтролерах ESP. Вона забезпечує створення Wi-Fi точки доступу (AP) або підключення до існуючої Wi-Fi мережі (STA), а також дозволяє виконувати оновлення прошивки OTA (Over-the-Air).

OTA оновлення — це процес оновлення програмного забезпечення або прошивки пристрою через бездротове з'єднання, зазвичай через Wi-Fi або мобільну мережу. OTA оновлення дозволяють розробникам випускати оновлення для своїх пристроїв без необхідності фізичного підключення до них, що значно спрощує процес підтримки та вдосконалення пристроїв.

На рисунку 3.7 зображено структурну схему роботи бібліотеки інтерфейсу користувача.

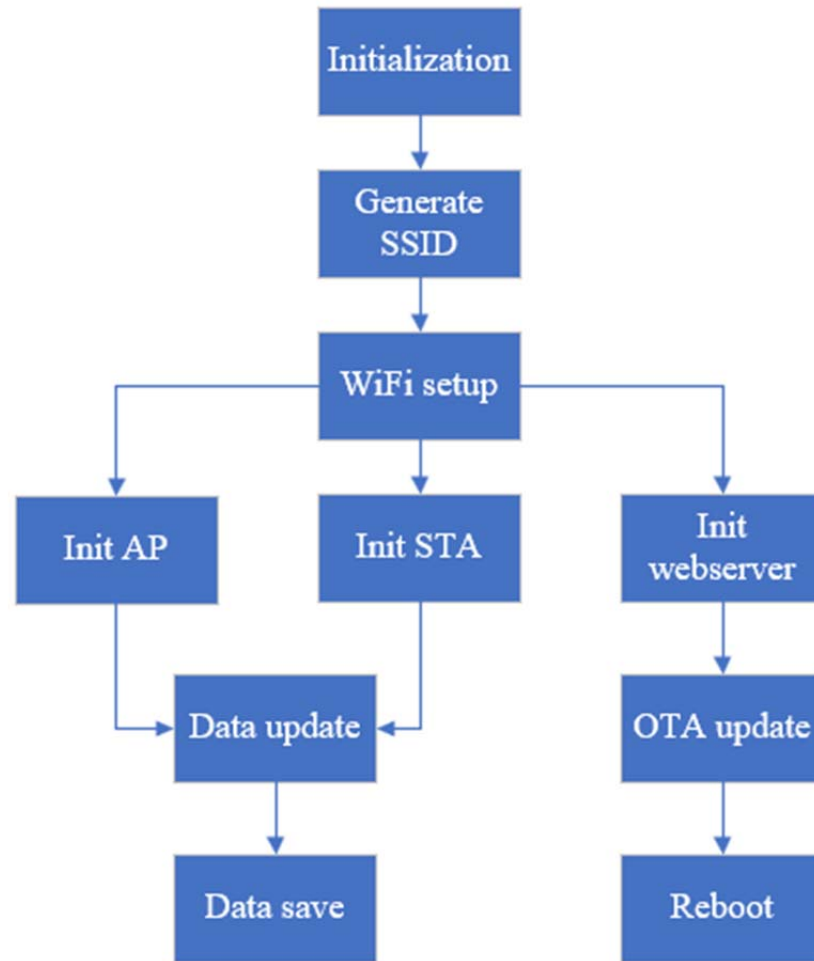


Рисунок 3.7 Структурна схема роботи бібліотеки інтерфейсу користувача.

На першому етапі, при створенні точки доступу (AP), викликається функцію “init\_wifi\_AP”, який ініціалізує модуль Wi-Fi, встановлює необхідні параметри мережі, такі як SSID, пароль, IP-адресу, шлюз та маску підмережі. Функція “num\_wifi” генерує унікальний SSID для точки доступу на основі ідентифікатора мікросхеми. Після цього створюється Wi-Fi точка доступу з викликом WiFi.softAPConfig та WiFi.softAP. Функція також запускає веб-сервер на порту 80 для обробки HTTP-запитів.

Якщо потрібно підключитися до існуючої Wi-Fi мережі, викликається функцію “init\_wifi\_STA”. Він налаштовує модуль Wi-Fi на режим станції (STA), використовує збережені налаштування SSID та

пароля, і виконує підключення до мережі через “WiFi.begin”. При успішному підключенні відображається локальна IP-адреса пристрою.

Функція “init\_web\_server” запускає веб-сервер та налаштовує DNS-відповідач для роботи з мережею. Він обробляє HTTP-запити на головну сторінку та на маршрут для оновлення прошивки “/update”. Під час процесу оновлення прошивки використовуються функції “Update”, що забезпечують прийом та запис нової прошивки в пам'ять мікроконтролера.

Функція “wifi\_setup” сканує наявні Wi-Fi мережі та вирішує, чи створювати точку доступу, чи підключатися до існуючої мережі на основі певних умов. Він також зчитує налаштування з енергонезалежної пам'яті за допомогою функції “read\_preferences”, яка отримує збережені значення чутливості та параметрів каналів для різних режимів (крило та квадрокоптер).

Функція “wifi\_loop” обробляє запити від підключених клієнтів. Він розпізнає різні HTTP-запити, такі як запити на сторінки налаштувань, збереження нових налаштувань або подачу форми з текстовим введенням. У разі запиту на збереження налаштувань викликаються функції “handleSaveSettingsWing” та “handleSaveSettingsQuadrocopter”, які зберігають нові значення чутливості та параметрів каналів у пам'яті.

Таким чином, бібліотека забезпечує повний функціонал для налаштування і керування Wi-Fi з'єднанням, а також для віддаленого оновлення прошивки пристрою через веб-сервер.

### 3.5 Протоколи роботи з зовнішніми модулями

Як видно з рисунку 3.1 польотний контролер в своїй роботі використовує різні протоколи для вхідних сигналів та вихідних сигналів. До вхідних сигналів належать сигнали керування БПЛА від приймача. Зачасту використовують приймачі з протоколом ELRS/CRSF.

Для керування зовнішніми модулями використовують сигнали PWM, DShot, SmartAudio/Tramp, MAVLink.

### 3.5.1 CRSF

CRSF (Crossfire) - це протокол зв'язку, розроблений для передачі даних з низькою затримкою та високою надійністю, який використовується переважно в дистанційних системах керування (RC) для БПЛА. Цей протокол дозволяє здійснювати керування дроном і передавати телеметрію на великі відстані з мінімальною затримкою, забезпечуючи стабільний і безперервний зв'язок між пілотом і дроном.

Протокол CRSF працює через серійний порт і підтримує швидкість передачі даних до 400 кбіт/с. Він оптимізований для використання з радіомодулями TBS Crossfire, що забезпечують дальність зв'язку до 100 км і більше, залежно від умов навколишнього середовища і потужності передавача.

Протокол використовує технологію FHSS (Frequency Hopping Spread Spectrum) для забезпечення стійкості до перешкод і збереження стабільного зв'язку в умовах високого рівня електромагнітних завад.

На рисунку 3.8 зображено структурну схему роботи бібліотеки `crsfSerial`.

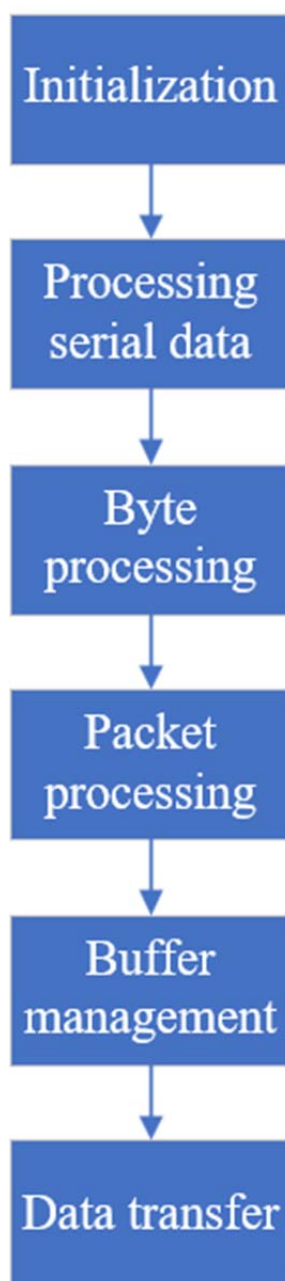


Рисунок 3.8 Структурна схема роботи бібліотеки crsfSerial

Детальніше розберемо її роботу. На першому етапі ініціалізації об'єкта CrsfSerial викликається конструктор, який встановлює початкові значення для послідовного порту, CRC, швидкості передачі даних та інших параметрів. Функція “begin” встановлює швидкість передачі даних для порту, якщо вона не дорівнює нулю. Якщо значення не задано, використовується значення за замовчуванням, задане під час створення об'єкта. Функція “handleSerialIn” зчитує байти з порту та обробляє їх.

Якщо ввімкнений режим passthrough, дані передаються безпосередньо на вихід. Інакше, байти зберігаються у буфері “\_rxBuf” і передаються для подальшої обробки функцією “handleByteReceived”. Якщо буфер переповнюється, він очищується для прийому нових даних. Функція “handleByteReceived” обробляє прийняті байти, перевіряючи цілісність пакету за допомогою контрольної суми (CRC). Якщо пакет валідний, він обробляється відповідною функцією, а буфер очищується для прийому наступного пакету. Якщо ж пакет некоректний, перший байт видаляється з буфера, і процес повторюється. Функція “checkPacketTimeout” перевіряє, чи не пройшов заданий час очікування пакету без отримання нових даних. Якщо так, буфер очищується по одному байту, щоб звільнити місце для нових даних. Функція “checkLinkDown” перевіряє стан з'єднання. Якщо з'єднання було активним, але протягом заданого часу не отримано нових пакетів, викликається обробник “onLinkDown”, з'єднання вважається втраченим.

### 3.5.2 PWM

У сучасних RC сервоприводах кут механічного повороту визначається шириною електричного імпульсу, що подається на керуючий провід. Це є формою широтно-імпульсної модуляції (ШІМ). Зазвичай RC сервоприводи очікують імпульс кожні 20 мс, але цей інтервал може значно варіюватися в залежності від моделі. Ширина імпульсу визначає, наскільки обертається двигун. Наприклад, у багатьох RC сервоприводах імпульс тривалістю 1,5 мс змушує двигун повертатися на 90° (нейтральне положення). Часовий інтервал між імпульсами може змінюватися без впливу на положення сервомотора. На рисунку 3.9 зображено діаграму, що показує типову синхронізацію ШІМ для сервомотора.

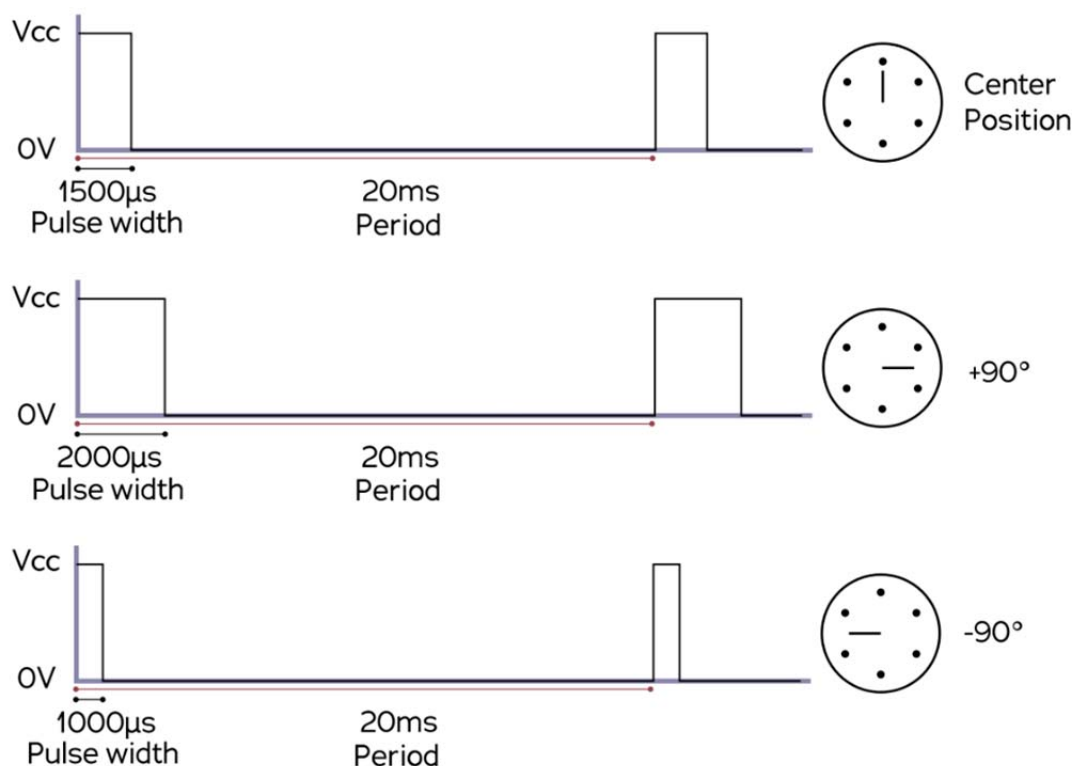


Рисунок 3.9 Діаграма, що показує типову синхронізацію ШІМ для сервомотора

Положення сучасного RC сервоприводу визначається виключно шириною імпульсу, а не робочим циклом ШІМ (часом увімкнення та вимкнення). Це відрізняється від ШІМ, який використовується, наприклад, у деяких регуляторах швидкості двигуна постійного струму. Більшість RC сервоприводів займають однакове положення при отриманні імпульсу 1,5 мс кожні 6 мс (завдання 25%) або кожні 25 мс (завдання 6%) – в обох випадках вони повертаються у центральне (нейтральне) положення. Якщо частота оновлення (кількість імпульсів за секунду) знаходиться в межах від 40 Гц до 200 Гц, точне значення частоти не має значення для більшості RC сервоприводів.

Період у 20 мс (50 Гц) походить з часів, коли сигнал передавався у форматі PPM (імпульсно-позиційна модуляція) по повітрю. Тоді період PPM складав близько 22,5 мс, і перетворення в ШІМ було простим: час

високого стану ШІМ відповідав часовому положенню імпульсу PPM для цього сервоприводу.

Більшість RC приймачів надсилають імпульси до сервоприводу з певною постійною частотою кадрів, змінюючи лише максимальний час. Однак, можна керувати рухом сервоприводу по всьому діапазону за допомогою генератора функцій, встановивши постійний робочий цикл 10% і змінюючи лише частоту (частоту кадрів).

### 3.5.3 Dshot

DShot - це сучасний протокол передачі даних між польотним контролером і ESC, який замінює старіші протоколи, такі як Oneshot і Multishot. Цей протокол був розроблений компанією Flyduino у співпраці з Betaflight.

Аналогові протоколи ШІМ, наприклад, Multishot, Oneshot42, Oneshot125 та інші, використовують ширину імпульсу для визначення положення дросельної заслонки, що має ряд значних недоліків. Одним з них є явище, відоме як "jitter", коли сигнал коливається навколо необхідної точки. Чим коротша ширина імпульсу, тим більша ймовірність і вплив цього тремтіння. Варіації ширини імпульсу також призводять до неточностей в нульовому та максимальному положеннях дроселя, через що потрібне калібрування ESC.

Більшість цифрових комунікацій використовують рівні синхронізації та напруги відносно певного еталона (зазвичай заземлення) для позначення значень бітів. Наприклад, у послідовному зв'язку (RS232, RS485 тощо) високий рівень напруги ( $>$  еталонного) означає біт 1, а низький рівень (наближений до еталонного або  $<$  еталонного) означає біт 0. Для послідовного зв'язку кожен кінець повинен знати синхронізацію, оскільки послідовні біти у стані 1 просто підтримують високий рівень без імпульсів, тому швидкість передачі є критично важливою.

У випадку з DShot часовий інтервал є менш важливим, оскільки значення біта визначається шириною імпульсу у відсотках від загального періоду. Це схоже на принцип роботи світлодіодних стрічок WS2811/2, за винятком відмінностей у таймінгах, вони майже ідентичні.

Протокол DSHOT складається з передачі 16-бітових пакетів до ESC: 11-бітового значення дроселя, 1-біта для запиту телеметрії та 4-бітової контрольної суми. Існує три основні швидкості протоколу: DShot150, DShot300 і DShot600. Також, завдяки використанню мікроконтролера ESP32 можливо реалізувати роботу новітнього протоколу DShot1200.

На рисунку 3.10 зображено таймінги різних швидкостей протоколу.

DSHOT	Bitrate	TH1	TH0	Bit Time $\mu$ s	Frame Time $\mu$ s
150	150kbit/s	5.00	2.50	6.67	106.72
300	300kbit/s	2.50	1.25	3.33	53.28
600	600kbit/s	1.25	0.625	1.67	26.72
1200	1200kbit/s	0.625	0.313	0.83	13.28

Рисунок 3.10 таймінги різних швидкостей протоколу.

З рисунку видно основні параметри сигналу протоколу DShot різних частот.

На рисунку 3.11 зображено принципову схему роботи бібліотеки DShotRMT.

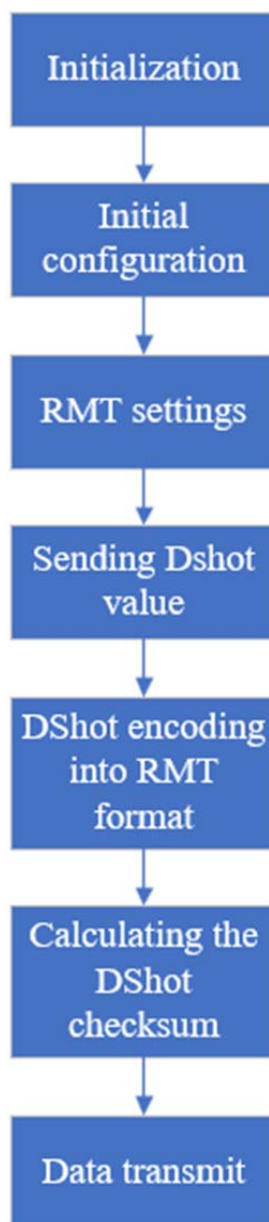


Рисунок 3.11 Принципова схема роботи бібліотеки DShotRMT.

Детальніше розберемо її роботу. На першому етапі ініціалізації об'єкта класу DShotRMT використовується один із конструкторів, що приймають номер GPIO і канал RMT або номер піну і канал. Конструктор налаштовує необхідні параметри, такі як номер GPIO, номер піну, канал RMT та кількість блоків пам'яті. Функція “begin” здійснює початкове налаштування, встановлює режим DShot (DShot150, DShot300, DShot600, DShot1200) і визначає, чи використовується двонаправлений режим. Залежно від режиму DShot, налаштовуються тривалість сигналів та інші

параметри. В кінці функції викликається функція налаштування RMT і встановлення драйвера RMT. Функція “send\_dshot\_value” відправляє значення для керування швидкістю двигуна (throttle\_value) і запит телеметрії (telemetric\_request). Спочатку обмежує значення throttle\_value до допустимого діапазону, потім формує пакет і передає його через RMT. Функція “encode\_dshot\_to\_rmt” кодує значення DShot у формат RMT. Залежно від того, чи активований двонаправлений режим, він налаштовує тривалість сигналів та їх рівні. Функція “calc\_dshot\_chksum” обчислює контрольну суму для пакету DShot, що необхідна для коректної передачі даних. Функція “prepare\_rmt\_data” готує пакет для передачі, включаючи обчислення та додавання контрольної суми.

#### 3.5.4 SmartAudio/Tramp

Для передачі відеосигналу від БПЛА до пілота використовують зовнішні модулі VTX. Найбільш поширеними на сьогодні є Rush Tank, АКК FX3, ImmersionRC Tramp, TBS Unify Pro, iFlight BLITZ. Всі вони мають можливість керування по UART. При цьому використовується два протоколи керування, а саме: TBS SmartAudio та IRC Tramp.

TBS SmartAudio - це протокол зв'язку між польотним контролером і відеопередавачем, що був винайдений компанією TBS у 2015 році. Він називається SmartAudio, тому що спочатку був призначений для спільного використання входу аудіосигналу VTX разом з лінією управління від польотного контролера. Пізніше SmartAudio отримав окремий виділений вхідний контакт у VTX, але слово Audio залишилося в назві.

IRC Tramp — це протокол зв'язку між польотним контролером та відеопередавачем БПЛА. Завдяки цьому протоколу контролер польоту може налаштовувати параметри відеопередавача такі як: потужність передачі, діапазон і канал відеосигналу.

SmartAudio і Tramp — це два різних протоколи, які дозволяють змінювати налаштування VTX. SmartAudio використовується в продуктах TBS (Team BlackSheep), а Tramp — в продуктах IRC (ImmersionRC). Обидва протоколи дозволяють змінювати параметри VTX за допомогою пульта дистанційного керування або через програмне забезпечення.

SmartAudio і Tramp фактично протоколи, реалізовані на однопровідному UART, тобто у напівдуплексному режимі.

Таким чином, для функціонування бібліотек з протоколами SmartAudio і Tramp необхідно в мікроконтролері ESP32 організувати роботу режиму HalfDuplex для відповідного UART. Це виконується наступним чином:

```
#define UART_PIN 17 – у бібліотеці зі змінними вказуємо адресу пінa
HardwareSerial VTX_port(1); - ініціалізуємо об'єкт VTX_port класу
HardwareSerial та використовуємо апаратний UART 1
```

```
VTX_port.begin(115200, SERIAL_8N1, UART_PIN, UART_PIN); -
запускаємо VTX_port з відповідними характеристиками.
```

Сворюємо дві функції які переназначають режим пінa режим пінa

```
void switchToTX() {
    pinMode(UART_PIN, OUTPUT);
}
```

```
void switchToRX() {
    pinMode(UART_PIN, INPUT);
}
```

В циклі здійснюємо запис команд у VTX та обробку відповіді.

```
void loop() {
    switchToTX();
    VTX_port.write("Data");
    VTX_port.flush();
    switchToRX();
    if (VTX_port.available()) {
```

```
String received =VTX_port.readString();  
    }  
}
```

У використаній бібліотеці визначаються параметри для налаштування відповідно до типу трансмітера, наприклад, потужність передавача (“setPowerInmW”) та частоти (“setFrequency”). Крім того, програма включає різні функції для налаштування каналів (“setChannel”, “setPrevChannel”, “setNextChannel”).

Також у кодї є реалізація функцій для взаємодії з двома різними протоколами: SmartAudio (“sa\_updateSettings”) та Tramp (“trampUpdate”). Кожен протокол має свої власні особливості та специфікації, які програма враховує при взаємодії з трансмітером.

### 3.5.5 OSD

OSD для БПЛА є важливою технологією, що дозволяє пілотам отримувати в реальному часі критичну інформацію про стан і параметри польоту безпосередньо на екрані відеопередавача або FPV окулярів. Ця інформація є надзвичайно корисною для забезпечення безпеки та ефективності польоту, а також для виконання конкретних завдань.

У контексті плат OSD, MAVLink використовується для передавання телеметричних даних, таких як інформація про політ, стан системи та параметри місії, які потім відображаються на екрані користувача. Це може включати дані про висоту, швидкість, курс, рівень заряду акумулятора, час польоту, GPS-координати, інформації про відстань до точки старту напрямку та інші важливі параметри польоту.

Найбільш популярні OSD модулі для БПЛА:

- MinimOSD: Один із найбільш популярних модулів, сумісний з багатьма польотними контролерами і забезпечує широкий спектр функцій.
- Betaflight OSD: Вбудований в багато сучасних контролерів, дозволяє легко налаштовувати і використовувати OSD прямо з конфігураційного інтерфейсу Betaflight.
- MWOSD: Багатофункціональний OSD, який підтримує різні польотні контролери і пропонує велику кількість налаштувань.

## РОЗДІЛ 4 ТЕСТУВАННЯ, РОЗРОБЛЕННЯ МЕТОДИКИ ЗАСТОСУВАННЯ

### 4.1 Тестування дослідного зразка польотного контролера

Дослідний зразок удосконаленої електронної системи керування БПЛА успішно пройшов тестування за допомогою створеного тестового програмного забезпечення в середовищі Visual Studio Code з використанням PlatformIO. Під час тестування було підтверджено, що всі компоненти схеми функціонують справно:

- Сенсори MPU6050, QMC5883L та BMP280 забезпечують стабільну передачу даних по шині I2C у відповідь на запити від ESP32. Це дозволяє точно вимірювати параметри руху, магнітне поле та атмосферний тиск відповідно.
- ESP32 програмується без проблем через середовище розробки та виконує свої функції, надаючи відповідну інформацію в процесі роботи, що включає збір даних з сенсорів, управління двигунами та комунікацію з іншими компонентами системи.

Крім цього, за допомогою осцилографа C1-67 було підтверджено якість реалізації принципової схеми.

Блок живлення ефективно перетворює 5В від контролера двигуна в стабільні 3.3В. Було виявлено, що електромагнітні завади на шинах живлення та заземлення повністю відсутні на стороні 3.3В, що свідчить про удосконалення завадостійкості польотного контролера. На рисунку 4.1 зображено осцилограму шини живлення польотного контролера Накс F722 з виключеними двигунами.

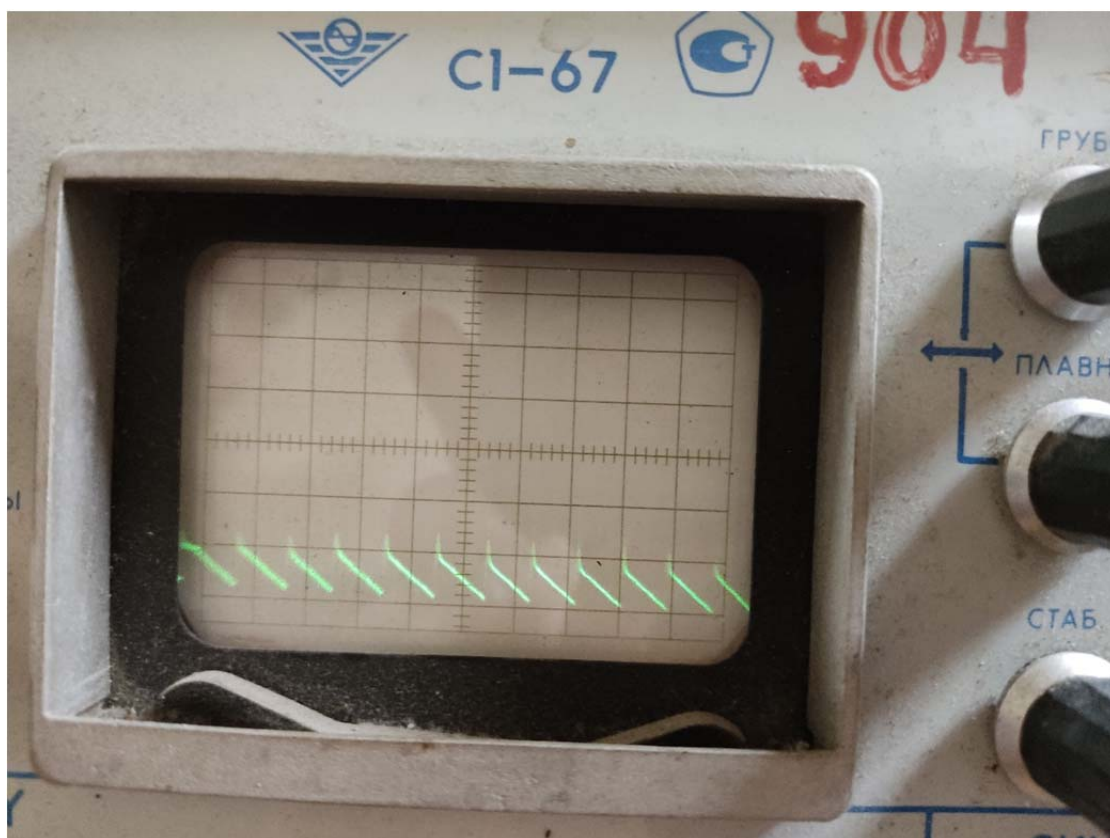


Рисунок 4.1 Осцилограма шини живлення польотного контролеру Накрс F722 з виключеними двигунами.

З рисунку видно, що на шині живлення польотного контролеру після включення з'являється систематична завада частотою 1кГц з амплітудою 20мВ. На рисунку 4.2 зображено осцилограму шини живлення польотного контролеру Накрс F722 з включеними двигунами.

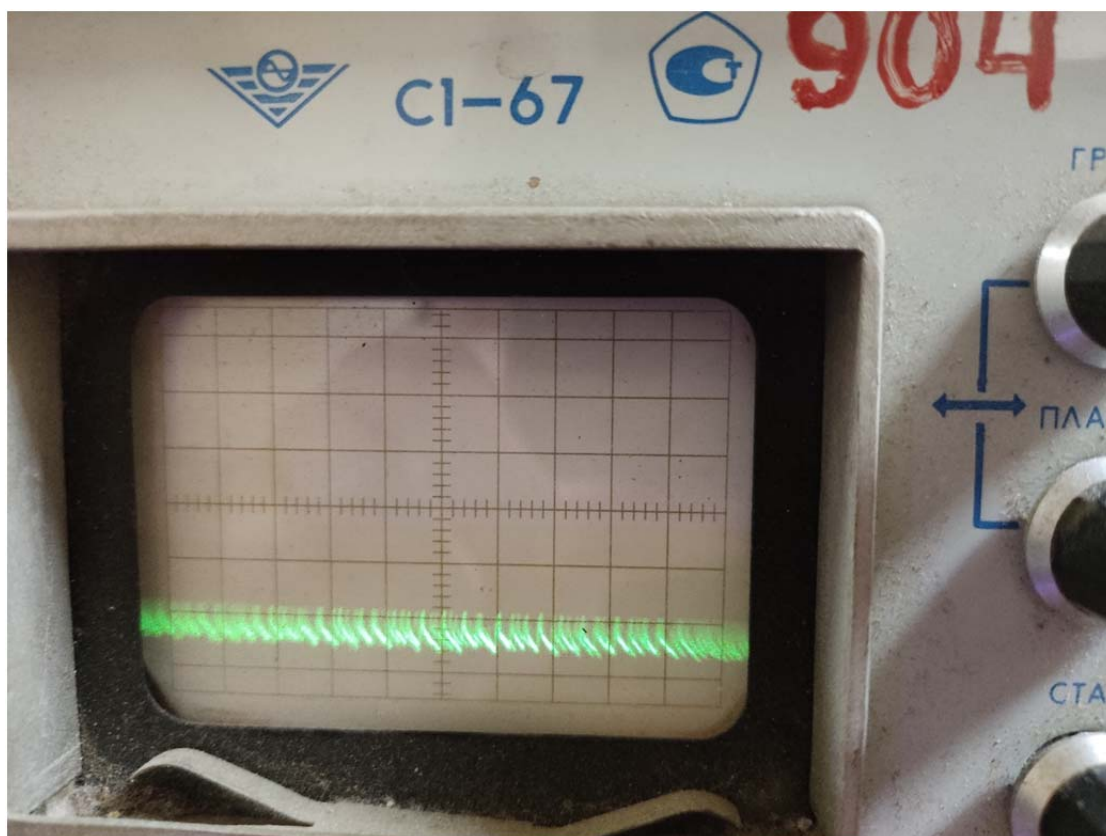


Рисунок 4.2 Осцилограма шини живлення польотного контролеру Накрс F722 з включеними двигунами.

З рисунку видно, що на шині живлення польотного контролеру після включення двигунів з'являється систематична завада частотою 8кГц з амплітудою 15мВ.

Натомість, спроектована схема удосконаленої електронної системи керування БПЛА позбавлена такого недоліку. На рисунку 4.3 зображено осцилограму шини живлення розробленого польотного контролера при включених двигунах.



Рисунок 4.3 осцилограма шини живлення розробленого польотного контролера при включених двигунах.

Як видно з рисунку 4.3, спроектована та реалізована система фільтрації повністю унеможлиблює появу як низькочастотної так і високочастотної завади.

Перетворювачі рівнів на основі польових транзисторів BSS138 надійно перетворюють логічні рівні сигналів з 3.3В, які використовує ESP32, до 5В, необхідних для контролера двигуна або сервоприводів. Це забезпечує сумісність інтерфейсів та надійність обміну даними. На рисунку 4.4 зображено осцилограми PWM сигналу для керування сервоприводом або ESC.



Рисунок 4.4 осцилограми PWM сигналу для керування сервоприводом або ESC.

Як видно з рисунку 4.4, амплітуда PWM сигналу дорівнює 5В. Передній та задній фронт прямокутного імпульсу сигналу PWM вертикальні, що є великою перевагою при керуванні потужними сервоприводами. Завади в сигналі PWM відсутні.

#### 4.2 Методика застосування

Польотний контролер – це ключовий компонент будь-якого БПЛА, який відповідає за стабілізацію та управління польотом. Це друкована плата, на яку нанесені радіоелектронні компоненти та графічні символи для спрощення монтажу та налаштування. Одним з таких символів є біла стрілка, яка вказує напрямок осі roll. Ця стрілка має співпадати з віссю roll БПЛА, щоб забезпечити коректну орієнтацію контролера в просторі.

Контролер кріпиться до рами БПЛА за допомогою монтажних отворів, які зазвичай розташовані в кутах плати. Для зменшення передачі вібрацій від двигунів та інших рухомих частин БПЛА, використовуються спеціальні гумові демпфери. Вони діють як амортизатори, поглинаючи вібрації та забезпечуючи більш стабільну роботу сенсорів.

У центрі плати розташований сенсор MPU6050, який є комбінованим акселерометром та гіроскопом. Він вимірює прискорення та кутові

швидкості по трьох осях, що дозволяє контролеру точно відстежувати положення БПЛА в просторі. Розташування MPU6050 в центрі симетрії БПЛА є критично важливим для забезпечення точності показань та ефективності стабілізації.

Неправильне розташування польотного контролера може призвести до некоректної роботи системи стабілізації, що в свою чергу може спричинити втрату контролю над БПЛА. Тому важливо дотримуватися інструкцій та переконатися, що всі компоненти встановлені правильно та надійно закріплені.

Завдяки точному монтажу та калібруванню, польотний контролер забезпечує стабільний та контрольований політ, дозволяючи оператору БПЛА зосередитися на виконанні завдань, а не на утриманні апарату в повітрі.

У нижній частині польотного контролера розташовані виходи PWM які можуть використовуватись відповідно до внесених налаштувань. Наприклад, при використанні польотного контролера для керування БПЛА літакового типу з системою скиду вантажу та камери, що обертається підключення до вихідних контактів буде наступне:

- група контактів 1 – елерони;
- група контактів 2 – елеватор;
- група контактів 3 – двигун;
- група контактів 4 – руль напрямку;
- група контактів 5 та 6 – система скиду вантажу;
- група контактів 7 та 8 – система обертання камери.

Горизонтальна група контактів, розташована праворуч, призначена для підключення приймача, який отримує сигнали від пульта дистанційного керування або автопілота.

При використанні польотного контролеру для керування БПЛА мультироторного типу групи контрактів 1-4 підключаються до ESC двигунів, решта використовується на розсуд пілота.

Група контрактів, що розташовані праворуч від ESP32 призначені для програмування польотного контролеру та вибору режимів:

- група B (boot) використовується для переведення ESP32 в режим програмування;
- група R (reconfiguration) використовується для переведення польотного контролеру в режим налаштування вхідними, вихідними та проміжними параметрами;
- група 1 використовується для активації режиму оновлення OTA;
- група 2 є резервною та розрахована на використання при подальшому оновленні програмного забезпечення.

Вибір режимів польоту та налаштування польотного контролера виконуються через інтерфейс веб-сторінки, який надає користувачеві гнучкість та контроль над різними параметрами.

Основні параметри, які можна налаштувати на веб-сторінці:

- Вибір режимів польоту. Користувач може вибрати з різних режимів польоту, таких як крило, літак, мультиротор.
- Налаштування чутливості. Параметри чутливості для органів керування пульта та виконавчих елементів можна налаштувати, щоб забезпечити більш точне та комфортне управління БПЛА.
- Функціональне призначення вихідних груп. Користувач може визначити, які функції будуть призначені кожній групі PWM виходів, наприклад, для керування сервоприводами, двигунами або іншими модулями.

- Протоколи комунікації. Встановлення протоколів для кожної вихідної групи, таких як PWM, DShot, SmartAudio/Tramp, залежно від сумісності з обладнанням.
- Оновлення програмного забезпечення. Перехід на сторінку OTA оновлення дозволяє користувачеві оновити програмне забезпечення польотного контролера безпосередньо через інтернет, що забезпечує доступ до останніх функцій та виправлень помилок.

Ці налаштування дозволяють оптимізувати роботу БПЛА та адаптувати його до конкретних завдань, умов польоту, додаткового обладнання

## ВИСНОВКИ

У ході виконання цієї дипломної роботи було виконане поставлене технічне завдання яке полягало у створенні удосконаленого прототипу системи керування БПЛА з підвищеною завадостійкістю плати польотного контролера та оптимізацією оброблення даних за допомогою цифрової фільтрації. Розроблений прототип продемонстрував коректну стабілізацію та управління, що є критично важливим для безпечного та ефективного використання БПЛА в різних умовах.

Основні досягнення роботи:

- Створення гнучкої архітектури. Розроблена модульна структура польотного контролера дозволяє легко адаптувати його до різних типів БПЛА та місій.
- Інтеграція сучасних сенсорів. Використання сучасних сенсорів, таких як MPU6050, QMC5883L, забезпечило високу точність вимірювань.
- Автоматизація калібрування. Розроблено метод автоматичного калібрування сенсорів для спрощення підготовки БПЛА до польоту.
- Розробка інтуїтивного інтерфейсу. Веб-інтерфейс для налаштування контролера зробив процес конфігурації БПЛА доступним навіть для користувачів без глибоких технічних знань та з мінімально необхідними технічними засобами.
- Покращення алгоритмів стабілізації. Алгоритми стабілізації були оптимізовані та доопрацьовані для забезпечення стабільного польоту навіть при різних погодних умовах, а реалізація схемотехнічного рішення у вигляді PCB дизайну дала можливість значно знизити рівень шумів (завад) від інших компонентів БПЛА.

Виконана робота з розробки схемотехнічного рішення та програмних алгоритмів для управління польотом БПЛА відповідає сучасним вимогам

галузі. Проте, існує пропозиція до розширення існуючого функціоналу. Зокрема, це стосується інтеграції додаткових комунікаційних протоколів, що забезпечать сумісність з широким спектром новітніх зовнішніх модулів, також пропонується розробка передового алгоритму для обробки цифрового відеопотоку, який, завдяки застосуванню технологій штучного інтелекту, дозволить здійснювати автономні польоти та виконувати складні місії з високим ступенем точності.

## ВИКОРИСТАНІ ДЖЕРЕЛА

1. Philips Semiconductors, The I2C-bus and how to use it, 1995 URL: [https://www.i2c-bus.org/fileadmin/ftp/i2c\\_bus\\_specification\\_1995.pdf](https://www.i2c-bus.org/fileadmin/ftp/i2c_bus_specification_1995.pdf)
2. Espressif Systems, ESP32-WROOM-32 Datasheet, 2023. URL: [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf)
3. InvenSense, MPU-6000 and MPU-6050 Product Specification, 2013 URL: <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>
4. QST Corporation, 3-Axis Magnetic Sensor QMC5883L, 2016 URL: <https://github.com/e-Gizmo/QMC5883L-GY-271-Compass-module/blob/master/QMC5883L%20Datasheet%201.0%20.pdf>
5. Bosh, BMP280 Digital Pressure Sensor Data sheet, 2015 URL: <https://cdn-shop.adafruit.com/datasheets/BST-BMP280-DS001-11.pdf>
6. Onsemi, N-Channel Logic Level Enhancement Mode Field Effect Transistor BSS138, 2005 URL: <https://www.onsemi.com/pdf/datasheet/bss138-d.pdf>
7. STMicroelectronics, Adjustable and fixed low drop positive voltage regulator, 2020 URL: <https://www.st.com/resource/en/datasheet/ld1117.pdf>
8. Freescale Semiconductor, Tilt Sensing Using a Three-Axis Accelerometer, 2013 URL: <https://www.nxp.com/docs/en/application-note/AN3461.pdf>
9. Freescale Semiconductor, Implementing a Tilt-Compensated eCompass using Accelerometer and Magnetometer Sensors, 2015 URL: <https://www.nxp.com/docs/en/application-note/AN4248.pdf>
10. Aircraft Rotations URL: <https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/aircraft-rotations/>
11. Robert W. Erickson, Dragan Maksimovic Fundamentals of Power Electronics, Springer, 2020 URL:

[https://elprivod.nmu.org.ua/files/converters/Robert\\_Erikson\\_fundamentals-of-power-electronics-3n\\_2020.pdf](https://elprivod.nmu.org.ua/files/converters/Robert_Erikson_fundamentals-of-power-electronics-3n_2020.pdf)

12. Zachariah Peterson, Рекомендації щодо формування структури шарів під час проектування високошвидкісних плат. *ChipNews*. 2023. №4. С.24-25. URL: <https://drive.google.com/file/d/16MmSfQFSbH2fi7rtwYKUQ7NAWKgT0KCL/view>